

**RANCANG BANGUN APLIKASI SURVEI KEPUASAN PENGGUNA  
BERBASIS *ANDROID* DENGAN FITUR KUSTOMISASI KUESIONER  
DAN *COHORT ANALYSIS* PADA SISTEM HELPDESK UP A TIK  
UNIVERSITAS LAMPUNG**

**(Skripsi)**

**Oleh**

**ANGGI DWI KURNIAWAN  
1915061021**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

**RANCANG BANGUN APLIKASI SURVEI KEPUASAN PENGGUNA  
BERBASIS *ANDROID* DENGAN FITUR KUSTOMISASI KUESIONER  
DAN *COHORT ANALYSIS* PADA SISTEM HELPDESK UP A TIK  
UNIVERSITAS LAMPUNG**

**Oleh**

**ANGGI DWI KURNIAWAN**

**Skripsi**

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

**Pada**

**Program Studi Teknik Informatika  
Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2026**

## ABSTRAK

### RANCANG BANGUN APLIKASI SURVEI KEPUASAN PENGGUNA BERBASIS ANDROID DENGAN FITUR KUSTOMISASI KUESIONER DAN COHORT ANALYSIS PADA SISTEM HELPDESK UPA TIK UNIVERSITAS LAMPUNG

Oleh

ANGGI DWI KURNIAWAN

Unit Penunjang Akademik (UPA) TIK Universitas Lampung belum memiliki mekanisme yang terstruktur untuk mengumpulkan dan menganalisis tingkat kepuasan pengguna terhadap layanan *helpdesk*. Penelitian ini bertujuan merancang dan mengembangkan sistem survei kepuasan pengguna berbasis *mobile* yang terintegrasi dengan *helpdesk* UPA TIK, meliputi pengelolaan kuesioner secara fleksibel berbasis tiket layanan, rekapitulasi hasil, dan analisis *cohort* kepuasan. Sistem dikembangkan menggunakan metodologi *Rapid and Participatory Application Development* (RAPD). Implementasi dilakukan dengan *framework Flutter* untuk aplikasi Android serta *web admin, Go (Golang)* untuk *backend API*, dan *PostgreSQL* sebagai basis data. Pada *Stage 2*, pengujian *blackbox* terhadap *API endpoint* menggunakan teknik *decision table* pada 45 kasus uji dari 10 *user story* dilaksanakan secara *iteratif* selama tujuh iterasi hingga seluruh 56 *assertion* lulus (100%). Pada *Stage 3*, pengujian *blackbox system test end-to-end* terhadap 38 skenario melalui antarmuka aplikasi menunjukkan bahwa seluruh skenario lolos pengujian. *User Acceptance Testing* dilakukan terhadap 27 responden menggunakan kuesioner skala *Likert 1–5* yang dikelompokkan berdasarkan *user story*. Hasil penilaian menunjukkan persentase sebesar 91,90% untuk pengguna umum, 95,00% untuk admin, dan 92,13% secara keseluruhan dengan kriteria penerimaan Sangat Baik. Hasil penelitian menunjukkan bahwa sistem mampu menyediakan mekanisme survei kepuasan yang terintegrasi dengan alur tiket layanan, mendukung pengelolaan *template* survei secara fleksibel, serta menyediakan analisis *cohort* untuk pemantauan kepuasan secara periodik. Berdasarkan hasil pengujian, sistem dinyatakan memenuhi kebutuhan fungsional dan layak digunakan.

Kata kunci: *cohort analysis, helpdesk, kustomisasi kuesioner, Rapid and Participatory Application Development, survei kepuasan pengguna.*

## **ABSTRACT**

### **DESIGN AND DEVELOPMENT OF AN ANDROID-BASED USER SATISFACTION SURVEY APPLICATION WITH CUSTOMIZED QUESTIONNAIRE AND COHORT ANALYSIS FEATURE AT THE ICT SUPPORT UNIT HELPDESK SYSTEM UNIVERSITY OF LAMPUNG**

**By**

**ANGGI DWI KURNIAWAN**

The Academic Support Unit (UPA) TIK at the University of Lampung lacks a structured mechanism to collect and analyze user satisfaction with its helpdesk services. This study aims to design and develop a mobile-based user satisfaction survey system integrated with the UPA TIK helpdesk, including customized questionnaire management based on service tickets, result recapitulation, and cohort satisfaction analysis. The system was developed using the Rapid and Participatory Application Development (RAPD) methodology. Implementation used the Flutter framework for Android applications and the web admin, Go (Golang) for the backend API, and PostgreSQL as the database. In Stage 2, black-box testing of API endpoints using decision table techniques was performed iteratively over 7 iterations on 45 test cases from 10 user stories, until all 56 assertions passed (100%). In Stage 3, end-to-end black-box testing of 38 scenarios via the application interface showed that all passed. User Acceptance Testing was conducted with 27 respondents using a Likert-scale questionnaire ranging from 1 to 5, grouped by user stories. The results showed 91.90% for general users, 95.00% for administrators, and 92.13% overall, meeting the Very Good acceptance criterion. The study results indicate that the system can provide an integrated satisfaction survey mechanism with the service ticket flow, support flexible survey template management, and offer cohort analysis for periodic satisfaction monitoring. Based on the test results, the system meets the functional requirements and is suitable for use.

**Keywords:** cohort analysis, customized questionnaire, helpdesk, Rapid and Participatory Application Development, user satisfaction survey.

Judul Skripsi : **RANCANG BANGUN APLIKASI SURVEI  
KEPUASAN PENGGUNA BERBASIS  
ANDROID DENGAN FITUR  
KUSTOMISASI KUESIONER DAN  
COHORT ANALYSIS PADA SISTEM  
HELPDESK UPA TIK UNIVERSITAS  
LAMPUNG**

Nama Mahasiswa : **Anggi Dwi Kurniawan**

Nomor Pokok Mahasiswa : 1915061021

Program Studi : Teknik Informatika

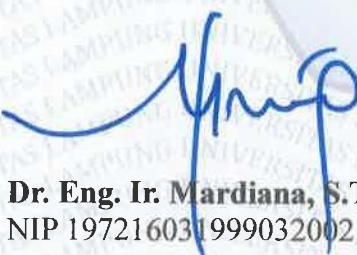
Fakultas : Teknik

### MENYETUJUI

#### 1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


  
**Dr. Eng. Ir. Mardiana, S.T., M.T., IPM.**  
NIP 197216031999032002


  
**Mahendra Pratama, S.T., M.Eng.**  
NIP 199112152019031013

#### 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

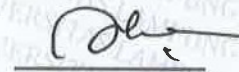
Ketua : **Dr. Eng. Ir. Mardiana, S.T, M.T., IPM.**



Sekretaris : **Mahendra Pratama, S.T., M.Eng.**



Penguji : **Dr. Ir. M. Komarudin, S.T., M.T.**



2. Dekan Fakultas Teknik



**Dr. Ahmad Herison, S.T., M.T.**  
NIP 196910302000031001

Tanggal Lulus Ujian Skripsi: 03 Juni 2026

## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Rancang Bangun Aplikasi Survei Kepuasan Pengguna Berbasis *Android* dengan Fitur Kustomisasi Kuesioner dan *Cohort Analysis* pada Sistem *Helpdesk* UPA TIK Universitas Lampung” 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 ketentuan akademik yang berlaku.

Bandar Lampung, 08 Juni 2026  
Pembuat pernyataan,



Anggi Dwi Kurniawan  
NPM 1915061021

## **RIWAYAT HIDUP**

Penulis dilahirkan di Purbolinggo, pada tanggal 17 April 2001. Penulis merupakan anak kedua dari tiga bersaudara, dari pasangan Bapak Sardi dan Ibu Sudarmi, serta dibesarkan dan dibina oleh orang tua penulis, yaitu Bapak Suwaji dan Ibu Suparti. Penulis memulai pendidikan formal di TK Dharma Wanita Mulya, pendidikan dasar di SDN 1 Bumi Dipasena Mulya hingga tahun 2011, dan dilanjutkan di SDN 2 Toto Harjo hingga tamat pada tahun 2013. Pendidikan menengah pertama di SMPN 1 Purbolinggo, dan menengah atas di SMAN 1 Purbolinggo. Pada tahun 2019, penulis melanjutkan pendidikan tinggi di Program Studi Teknik Informatika, Fakultas Teknik, Universitas Lampung (Unila), melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis aktif berorganisasi sebagai anggota pengurus Himpunan Mahasiswa Elektro (HIMATRO) Unila pada tahun 2020-2021. Selain itu, penulis juga meraih prestasi non akademik selama menjadi mahasiswa, yaitu Juara 1 Lomba Podcast Pembacaan Cerpen yang diadakan oleh Universitas Negeri Yogyakarta (UNY) pada tahun 2020, serta Juara Harapan 1 dan 2 pada lomba pembacaan cerpen yang juga diselenggarakan oleh UNY pada tahun 2021 dan 2022. Penulis melaksanakan Kuliah Kerja Nyata di Desa Kali Bening, Kecamatan Pekalongan, Kabupaten Lampung Timur, Provinsi Lampung pada tahun 2022. Penulis juga turut mengikuti kegiatan Kampus Merdeka selama dua semester, yakni program Magang Bersertifikat sebagai software engineer di PT Gamatechno Indonesia serta sebagai programmer di Badan Penyelenggara Jaminan Produk Halal (BPJPH) Kementerian Agama Republik Indonesia pada tahun 2022.

## MOTTO

“Dan katakanlah kebenaran itu datang dari Tuhanmu.”

*(QS. Al-Kahfi: 29)*

“Janganlah kamu bersikap lemah, dan jangan pula bersedih hati, padahal kamulah yang paling tinggi derajatnya jika kamu beriman.”

*(QS. Ali Imran: 139)*

“Barang siapa keluar untuk mencari sebuah ilmu, maka ia akan berada di jalan Allah hingga ia kembali.”

*(HR Tirmidzi)*

“Pengetahuan yang baik adalah yang memberikan manfaat, bukan hanya diingat.”

*(Imam Syafi’i)*

“It does not matter how slowly you go as long as you do not stop.”

*(Confucius)*

“Ketika kamu gagal dan berputus asa, ingatlah satu hal ‘apakah kamu akan menorehkan cerita buruk dalam setiap perjalananmu?’”

*(Penulis)*

“Success is not final, failure is not fatal: it is the courage to continue that counts.”

*(Winston Churchill)*

## PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

**Dengan mengharapakan Ridho dari Allah SWT,  
Alhamdulillah, Atas Izin Allah yang Maha Kuasa  
Kupersembahkan karya saya ini kepada:**

“Orang tua tercinta, yang tanpa lelah menghadirkan doa dalam diam, menguatkan dalam sunyi, dan mencintai tanpa syarat. Segala pencapaian ini tidak pernah benar-benar menjadi milik penulis seorang diri, melainkan juga buah dari kasih sayang dan pengorbanan mereka yang tidak terbalas oleh apa pun.”

“Keluarga besar penulis, yang menjadi tempat pulang paling hangat ketika dunia terasa melelahkan. Dukungan, perhatian, dan kebersamaan kalian menjadi salah satu alasan penulis mampu bertahan hingga titik ini.”

“Diri sendiri, yang telah berjuang keras, dengan segala keteguhan hati dan kekuatan. Setiap keringat, air mata yang menetes dalam perjalanan menjadikan perjuangan yang tidak sia-sia.”

“Teman-teman, Bapak Ibu Dosen, serta seluruh pihak yang selalu memberikan dukungan dan membantu selama perkuliahan hingga penulisan skripsi ini”

“Almamater tercinta, tempat penulis tumbuh, jatuh, belajar, dan perlahan menemukan bentuk diri yang lebih dewasa.”

## SANWACANA

Puji dan syukur penulis panjatkan ke hadirat Allah SWT, atas segala rahmat dan karunia-Nya karya tulis ini dapat terselesaikan. Selawat serta salam semoga senantiasa tercurahkan kepada Baginda Nabi Muhammad SAW, suri teladan sepanjang zaman, semoga kelak mendapatkan syafaatnya di yaumul kiamah nanti. Penelitian dengan judul “Rancang Bangun Aplikasi Survei Kepuasan Pengguna Berbasis *Android* dengan Fitur Kustomisasi Kuesioner dan *Cohort Analysis* pada Sistem *Helpdesk* UPA TIK Universitas Lampung” ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Di balik setiap halaman yang tersusun, terdapat bantuan, bimbingan, dan dukungan dari berbagai pihak yang tidak ternilai harganya. Oleh karena itu, dengan penuh ketulusan penulis mengucapkan terima kasih kepada:

1. Orang tua tercinta penulis, serta segenap keluarga penulis, yang senantiasa memberikan doa, dukungan, dan kasih sayang tanpa akhir; kehadiran dan motivasi mereka menjadi sumber kekuatan yang tak ternilai selama perjalanan akademik ini.
2. Bapak Dr. Ahmad Herison, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung, yang telah membantu penulis dalam menempuh proses akademik.
4. Ibu Yessi Mulyani, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung, serta sebagai Dosen Pembimbing Akademik yang telah memberikan arahan dan membantu dalam proses perkuliahan penulis.
5. Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., IPM. selaku Dosen Pembimbing Utama, yang dengan kesabaran meluangkan waktu untuk membimbing penulis dalam

penyusunan skripsi ini serta memberikan masukan dan nasihat yang sangat bermanfaat bagi peningkatan kualitas karya tulis ini..

6. Bapak Mahendra Pratama, S.T., M.Eng. selaku dosen pembimbing pendamping, yang bersedia memberikan bimbingan dan arahan teknis sehingga penulis dapat menyelesaikan penelitian dan penulisan skripsi dengan lebih terarah.
7. Bapak Dr. Ir. Komarudin, S.T., M.T. selaku dosen penguji, yang telah memberikan masukan dan saran yang membangun kepada penulis dalam mengerjakan skripsi ini.
8. Segenap Dosen di Jurusan Teknik Elektro yang telah menyampaikan ilmu, wawasan, dan pengalaman berharga selama masa perkuliahan.
9. Segenap tenaga kependidikan dan staf di Jurusan Teknik Elektro serta Fakultas Teknik yang telah membantu penulis dalam urusan administrasi dan berbagai kebutuhan akademik lainnya sehingga proses studi berjalan lancar.
10. Keluarga besar Teknik Elektro angkatan 2019 yang telah menjadi teman seperjuangan; dengan kebersamaan dan semangat yang menambah warna dalam masa studi penulis.
11. Segenap teman-teman penulis, Ahmad Ilham, Aldi, Arya, Aser, Bagus, Dina, Ellangga, serta seluruh teman-teman Teknik Informatika 2019 yang senantiasa tolong-menolong, terima kasih telah menjadi bagian dari lembaran penulis selama masa perkuliahan.
12. Tim KKN Kali Bening, rekan dalam pengabdian selama 40 hari, terima kasih atas kebersamaan, cerita dan pengalamannya.
13. Tim MSIB di PT Gamatechno Indonesia, terima kasih telah melengkapi perjalanan pembelajaran penulis selama menjalani program magang di Daerah Istimewa Yogyakarta.
14. Tim Divisi Programmer MSIB di BPJPH serta penghuni ruang kelas, terima kasih atas kebersamaan yang menemani perjalanan penulis, melengkapi cerita dan memberikan makna berharga selama menjalani program magang di Jakarta. Tidak lupa juga, keseharian bersama teman-teman penghuni Kost Aritonang yang selalu menghadirkan cerita baru setiap harinya.

Penulis menyadari bahwa karya ini masih membutuhkan ruang untuk berkembang dan diperbaiki. Segala kritik serta saran yang membangun akan diterima dengan lapang dada demi penyempurnaan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, institusi, dan para pembaca yang berkepentingan.

Bandar Lampung, 08 Juni 2026  
Penulis,

Anggi Dwi Kurniawan

## DAFTAR ISI

	Halaman
<b>DAFTAR TABEL .....</b>	<b>viii</b>
<b>DAFTAR GAMBAR .....</b>	<b>x</b>
<b>I. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Batasan Penelitian .....	5
1.6 Sistematika Penulisan.....	6
<b>II. TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1 Kepuasan Pengguna.....	7
2.2 Survei Kepuasan .....	8
2.3 <i>Helpdesk</i> .....	10
2.4 Sistem Informasi.....	11
2.5 <i>Cohort Analysis</i> .....	12
2.6 Teknologi Pengembangan Sistem.....	13
2.6.1 <i>Go</i> .....	13
2.6.2 <i>Android</i> .....	13
2.6.3 <i>Dart</i> .....	15
2.6.4 <i>Flutter</i> .....	16
2.6.5 <i>PostgreSQL</i> .....	17
2.6.6 <i>Firebase Cloud Messaging (FCM)</i> .....	18
2.7 Metode Pengembangan Sistem.....	19
2.7.1 <i>Rapid Application Development (RAD)</i> .....	19
2.7.2 <i>Participatory Design (PD)</i> .....	21
2.7.3 <i>Rapid and Participatory Application Development (RAPD)</i> .....	24
2.8 <i>Unified Modeling Language (UML)</i> .....	26
2.8.1 <i>Use Case Diagram</i> .....	27
2.8.2 <i>Activity Diagram</i> .....	27
2.8.3 <i>Class Diagram</i> .....	28
2.8.4 <i>Sequence Diagram</i> .....	29
2.9 Pengujian Perangkat Lunak.....	30
2.9.1 <i>White-box Testing</i> .....	30
2.9.2 <i>Unit Testing</i> .....	30
2.9.3 <i>Black-box Testing</i> .....	31
2.9.4 <i>Risk-Based Testing</i> .....	34

2.9.5	<i>User Acceptance Testing (UAT)</i> .....	35
2.10	Penelitian yang Berkaitan.....	39
2.10.1	<i>Unlocking User Satisfaction: A DeLone &amp; McLean IS Success Model Approach to IT Helpdesk Ticketing System Adoption</i> .....	39
2.10.2	<i>Analyzing User Satisfaction with the Helpdesk System of PT. Matahari Department Store: An Evaluation Study</i> .....	39
2.10.3	<i>Implementing online questionnaires and surveys by using mobile applications</i> .....	40
2.10.4	Penerapan <i>Waterfall</i> dalam Pengembangan Modul Survei Kepuasan Layanan pada Sistem Informasi Akademik Universitas Islam Negeri Antasari Banjarmasin.....	40
2.10.5	Sistem Informasi Survei Kepuasan Masyarakat dengan <i>Framework Laravel</i> pada Pelayanan Publik Pemerintah Daerah .....	40
2.10.6	<i>User Centered Method for Enhancing the Adoption of Software Systems in Palestine</i> .....	41
2.10.7	<i>RAPD: Rapid and Participatory Application Development of Usable Systems During COVID19 Crisis</i> .....	41
2.10.8	<i>A digital Cohort Analysis of Consumers' Mobile Banking App Experience</i> .....	42
2.10.9	<i>Risk-based Testing: Risk Analysis Fundamentals and Metrics for Software Testing Including a Financial Application Case Study</i> ..	42
<b>III.</b>	<b>METODOLOGI PENELITIAN .....</b>	<b>48</b>
3.1	Waktu dan Tempat .....	48
3.2	Alat dan Bahan .....	49
3.2.1	Alat Penelitian .....	49
3.2.2	Bahan Penelitian.....	50
3.3	Tahapan Penelitian.....	51
3.3.1	<i>Stage 1</i> .....	53
3.3.2	<i>Stage 2</i> .....	56
3.3.3	<i>Stage 3</i> .....	59
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>61</b>
4.1	Hasil.....	61
4.2	Implementasi .....	63
4.2.1	<i>Rapid and Participatory Application Development (RAPD)</i> .....	63
4.3	Pembahasan .....	183
4.3.1	Analisis Fungsionalitas Aplikasi .....	183
4.3.2	Analisis Hasil <i>User Acceptance Testing (UAT)</i> .....	185
4.3.3	Jawaban Atas Rumusan Masalah.....	187
<b>V.</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>190</b>
5.1	Kesimpulan.....	190
5.2	Saran .....	191
	<b>DAFTAR PUSTAKA.....</b>	<b>192</b>
	<b>LAMPIRAN.....</b>	<b>196</b>
	Lampiran 1 – <i>Source Code</i> Lengkap.....	196
	Lampiran 2 – Riwayat Perbaikan Fitur .....	197
	Lampiran 3 – Analisis Risiko <i>Pure Function</i> per Modul.....	198

Lampiran 4 – <i>Decision Matrix</i> dan <i>Test Case</i> per <i>User Story</i> .....	199
Lampiran 5 – Konfigurasi Lingkungan <i>Deployment (Staging)</i> .....	200
Lampiran 6 – Jawaban Kuesioner <i>User Acceptance Testing (UAT)</i> .....	201

## DAFTAR TABEL

	Halaman
Tabel 2.1 Komponen <i>Use Case Diagram</i> .....	27
Tabel 2.2 Komponen <i>Activity Diagram</i> .....	28
Tabel 2.3 Komponen <i>Class Diagram</i> .....	28
Tabel 2.4 Komponen <i>Sequence Diagram</i> .....	29
Tabel 2.5 Contoh Struktur <i>Decision Table</i> [14] .....	33
Tabel 2.6 Tahap Perhitungan Skor UAT.....	37
Tabel 2.7 Rentang Persentase Penerimaan.....	38
Tabel 2.8 Penelitian Terkait Kepuasan Pengguna pada sistem <i>helpdesk</i> .....	43
Tabel 2.9 Penelitian Terkait Survei <i>Online/Digital</i> .....	44
Tabel 2.10 Penelitian Terkait RAPD .....	45
Tabel 2.11 Penelitian Terkait <i>Cohort Analysis</i> .....	46
Tabel 2.12 Penelitian Terkait Risk-based Testing .....	46
Tabel 3.1 Jadwal Penelitian.....	48
Tabel 3.2 Perangkat Lunak.....	49
Tabel 3.3 Perangkat Keras.....	50
Tabel 3.4 Klasifikasi Risiko .....	57
Tabel 3.5 Spesifikasi Kebutuhan Server dan Layanan Deployment .....	58
Tabel 4.1 Prioritas persiapan kebutuhan pada <i>Stage 1</i> .....	63
Tabel 4.2 Prioritas pengembangan <i>prototype</i> fungsional pada <i>Stage 2</i> .....	63
Tabel 4.3 Prioritas penyempurnaan sistem dan <i>deployment</i> pada <i>Stage 3</i> .....	64
Tabel 4.4 <i>User Stories</i> dan <i>Acceptance Kriteria</i> .....	64
Tabel 4.5 Kebutuhan Fungsional Sistem.....	72
Tabel 4.6 Kebutuhan Non-Fungsional Sistem .....	73
Tabel 4.7 Analisis Risiko Per Modul.....	145
Tabel 4.8 <i>Hasil Unit Testing</i> Fungsi Murni.....	147
Tabel 4.9 Ringkasan Hasil <i>Unit Testing</i> per Modul .....	153
Tabel 4.10 Matriks <i>User Story</i> dan Jumlah Kasus Uji <i>Blackbox</i> .....	155
Tabel 4.11 Daftar Kegagalan Iterasi 1 .....	156
Tabel 4.12 Kegagalan yang Diperbaiki pada Iterasi 2 .....	158
Tabel 4.13 Kegagalan yang Diperbaiki pada Iterasi 3 .....	158
Tabel 4.14 Regresi dan Kegagalan Baru pada Iterasi 4 .....	159
Tabel 4.15 Kegagalan yang Diperbaiki pada Iterasi 6 .....	160
Tabel 4.16 Kegagalan yang Diperbaiki pada Iterasi 7 .....	160
Tabel 4.17 Rekapitulasi Hasil Seluruh Iterasi <i>Blackbox Testing</i> .....	161
Tabel 4.18 Pengujian <i>System</i> US01 .....	164
Tabel 4.19 Pengujian <i>System</i> US02.....	165
Tabel 4.20 Pengujian <i>System</i> US03 .....	166
Tabel 4.21 Pengujian <i>System</i> US04 .....	166
Tabel 4.22 Pengujian <i>System</i> US05.....	167

Tabel 4.23 Pengujian <i>System</i> US06.....	167
Tabel 4.24 Pengujian <i>System</i> US07.....	168
Tabel 4.25 Pengujian <i>System</i> US08.....	169
Tabel 4.26 Pengujian <i>System</i> US09.....	170
Tabel 4.27 Pengujian <i>System</i> US10.....	171
Tabel 4.28 Skenario UAT Pengguna Umum .....	172
Tabel 4.29 Skenario UAT Admin .....	173
Tabel 4.30 Hasil Penilaian UAT Pengguna Umum .....	173
Tabel 4.31 Hasil Penilaian UAT Admin .....	176
Tabel 4.32 Rekap Hasil UAT Keseluruhan .....	178
Tabel 4.33 Saran UAT dan Tindak Lanjut Perbaikan.....	182

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Fungsi Sistem Informasi [22].	11
Gambar 2.2 <i>Cohort Analysis</i> Retensi Pengguna.	12
Gambar 2.3 Arsitektur <i>Android</i> [25].	14
Gambar 2.4 Arsitektur <i>Flutter</i> [27].	16
Gambar 2.5 Siklus Hidup <i>Rapid Application Development (RAD)</i> [31].	20
Gambar 2.6 Generasi Ketiga <i>Participatory Design</i> [34].	23
Gambar 2.7 <i>Stage 1 RAPD</i> [12].	24
Gambar 2.8 <i>Stage 2 RAPD</i> [12].	25
Gambar 2.9 <i>Stage 2 RAPD</i> [12].	26
Gambar 2.10 Metode <i>Black-box Testing</i> [36].	31
Gambar 3.1 Tahapan Penelitian [11].	51
Gambar 4.1 Diagram Arsitektur Sistem.	74
Gambar 4.2 <i>Use Case Diagram</i> .	75
Gambar 4.3 <i>Activity Diagram Login</i> .	78
Gambar 4.4 <i>Activity Diagram Logout</i> .	79
Gambar 4.5 <i>Activity Diagram</i> Notifikasi <i>Push</i> .	80
Gambar 4.6 <i>Activity Diagram</i> Buat Tiket Layanan.	81
Gambar 4.7 <i>Activity Diagram</i> Buat Tiket Pengguna Tamu.	82
Gambar 4.8 <i>Activity Diagram</i> Mengisi Umpan Balik.	83
Gambar 4.9 <i>Activity Diagram</i> Melihat Riwayat Survei.	85
Gambar 4.10 <i>Activity Diagram</i> Pengelolaan Kuesioner oleh Admin.	86
Gambar 4.11 <i>Activity Diagram</i> Melihat Laporan Ringkasan Survei.	87
Gambar 4.12 <i>Activity Diagram</i> Melihat Analisis <i>Cohort</i> .	89
Gambar 4.13 <i>Activity Diagram</i> Melihat Riwayat Survei (Admin).	90
Gambar 4.14 <i>Sequence Diagram Login</i> .	91
Gambar 4.15 <i>Sequence Diagram Logout</i> .	92
Gambar 4.16 <i>Sequence Diagram</i> Notifikasi <i>Push</i> .	93
Gambar 4.17 <i>Sequence Diagram</i> Buat Tiket Layanan Pengguna Terdaftar.	95
Gambar 4.18 <i>Sequence Diagram</i> Buat Tiket Pengguna Tamu.	96
Gambar 4.19 <i>Sequence Diagram</i> Mengisi Umpan Balik.	97
Gambar 4.20 <i>Sequence Diagram</i> Melihat Riwayat Survei.	99
Gambar 4.21 <i>Sequence Diagram</i> Mengelola Kuesioner.	100
Gambar 4.22 <i>Sequence Diagram</i> Melihat Laporan Survei Kepuasan.	102
Gambar 4.23 <i>Sequence Diagram</i> Melihat Analisis <i>Cohort</i> .	103
Gambar 4.24 <i>Sequence Diagram</i> Melihat Riwayat Survei (Admin).	104
Gambar 4.25 <i>Class Diagram</i> .	105
Gambar 4.26 <i>Entity Relationship Diagram (ERD)</i> .	107
Gambar 4.27 Tampilan <i>Splash Screen</i> : (a) Aplikasi Mobile, (b) <i>Website</i> .	109
Gambar 4.28 <i>Code Snippet splash screen</i> .	110

Gambar 4.29 Tampilan Halaman <i>Login</i> : (a) Aplikasi Mobile, (b) <i>Website</i> .....	111
Gambar 4.30 <i>Code Snippet Handler Login</i> .....	112
Gambar 4.31 Tampilan Halaman <i>Form</i> Tiket Pengguna Tamu.....	113
Gambar 4.32 <i>Code Snippet</i> Halaman <i>Form</i> Tiket Pengguna Tamu. ....	114
Gambar 4.33 Tampilan Halaman Beranda ( <i>Home/Dashboard Page</i> ).....	115
Gambar 4.34 <i>Code Snippet List Ticket Handler</i> Berdasarkan <i>User</i> . ....	116
Gambar 4.35 Tampilan Halaman Daftar Tiket. ....	117
Gambar 4.36 <i>Code Snippet</i> Halaman Daftar Tiket.....	118
Gambar 4.37 Tampilan Halaman Pembuatan Tiket. ....	119
Gambar 4.38 <i>Code Snippet</i> Halaman Pembuatan Tiket. ....	120
Gambar 4.39 Tampilan Halaman Detail Tiket. ....	121
Gambar 4.40 <i>Code Snippet</i> Detail Tiket. ....	122
Gambar 4.41 Tampilan Halaman Umpan Balik: (a) Umpan balik Belum diisi, (b) Umpan Balik Sudah Diisi. ....	123
Gambar 4.42 <i>Code Snippet</i> Halaman Umpan Balik.....	124
Gambar 4.43 <i>Form</i> Survei Kepuasan.....	125
Gambar 4.44 <i>Code Snippet</i> Survei Kepuasan pada <i>Survey Handler</i> . ....	126
Gambar 4.45 <i>Code Snippet</i> Survei Kepuasan pada <i>Survey Service</i> . ....	127
Gambar 4.46 Tampilan Halaman Notifikasi. ....	128
Gambar 4.47 <i>Code Snippet Handler</i> Notifikasi. ....	129
Gambar 4.48 Tampilan Notifikasi <i>Push</i> pada Perangkat <i>Android</i> . ....	130
Gambar 4.49 <i>Code Snippet Ticket Service</i> untuk Notifikasi <i>Push</i> . ....	131
Gambar 4.50 Tampilan Halaman Profil. ....	132
Gambar 4.51 <i>Code Snippet Handler Logout</i> .....	133
Gambar 4.52 Tampilan Halaman Dasbor Admin. ....	134
Gambar 4.53 <i>Code Snippet</i> Dasbor Admin dari <i>Report Handler</i> .....	135
Gambar 4.54 Tampilan Halaman Laporan Survei.....	136
Gambar 4.55 <i>Code Snippet Report Handler</i> . ....	137
Gambar 4.56 Tampilan Halaman Analisis Kohort: (a) <i>Overview</i> , (b) <i>Heatmap</i> . ....	138
Gambar 4.57 <i>Code Snippet</i> Halaman Analisis Kohort.....	139
Gambar 4.58 Tampilan Halaman Pengaturan Survei: (a) Ringkasan, (b) Mengelola Kuesioner, (c) Atur <i>Template</i> pada Kategori. ....	140
Gambar 4.59 <i>Code Snippet Survey Handler</i> untuk <i>Template</i> Survei. ....	141
Gambar 4.60 <i>Code Snippet Category Handler</i> untuk <i>Template</i> Survei. ....	142
Gambar 4.61 Tampilan Halaman Riwayat Survei Admin.....	143
Gambar 4.62 <i>Code Snippet</i> Halaman Riwayat Survei Admin. ....	143
Gambar 4.63 Hasil Eksekusi <i>Unit Test</i> pada Terminal VS Code. ....	154
Gambar 4.64 Hasil <i>Run Collection</i> Menggunakan <i>Postman</i> pada Iterasi Akhir. ....	162
Gambar 4.65 Verifikasi server pengujian menggunakan <i>Postman</i> .....	163
Gambar 4.66 Pesan Kesalahan yang Diperbaharui: (a) Kesalahan <i>Login</i> , (b) Tidak Ada Koneksi Internet, (c) Kegagalan Pengiriman Kuesioner. ....	179
Gambar 4.67 <i>Code Snippet Input Validation</i> .....	180
Gambar 4.68 Halaman Riwayat Survei Admin yang Telah Diperbaiki. ....	181

# I. PENDAHULUAN

## 1.1 Latar Belakang

Pelayanan umum yang optimal merupakan salah satu elemen penting dalam mewujudkan tata kelola organisasi yang efektif. Kualitas pelayanan sangat dipengaruhi oleh bagaimana pengguna menilai pengalaman yang mereka terima, sehingga organisasi perlu mengukur kepuasan pengguna sebagai indikator keberhasilan layanan [1]. Di Indonesia, pengukuran kepuasan pengguna ditegaskan dalam UU No.25 tahun 2009 tentang Pelayanan Publik, yang menekankan evaluasi secara berkala sebagai dasar perbaikan prosedur, maupun penyempurnaan organisasi [2]. Salah satu pendekatan yang umum digunakan untuk memperoleh gambaran pengalaman pengguna ialah survei kepuasan. Melalui survei, penyelenggara layanan publik dapat memperoleh umpan balik pengguna yang dapat digunakan untuk menilai kualitas pelayanan dan mengidentifikasi area yang perlu ditingkatkan [3], [4].

Pada layanan berbasis sistem digital seperti *helpdesk*, kepuasan pengguna juga menjadi indikator penting dalam keberhasilan layanan digital karena pengalaman pengguna tidak hanya dipengaruhi aspek teknis, tetapi juga persepsi dan pengalaman selama berinteraksi dengan layanan [5], [6]. Meskipun *helpdesk* dinilai efektif untuk pencatatan keluhan dan penanganan tiket, organisasi membutuhkan mekanisme yang lebih sistematis untuk mengukur kepuasan pengguna setelah layanan diberikan. Pada praktiknya, pengukuran kepuasan masih dilakukan terpisah dari sistem operasionalnya, sehingga umpan balik tidak terkait langsung dengan tiket atau interaksi tertentu. Sebuah penelitian menunjukkan bahwa survei dapat dibangun sebagai modul tambahan pada sistem informasi yang telah ada [4].

Unit Penunjang Akademis Teknologi Informasi dan Komunikasi (UPA TIK) Universitas Lampung telah mengoperasikan sistem *helpdesk* berbasis web sebagai sarana utama untuk pelaporan insiden dan permintaan dukungan dari civitas *academica*. Namun, pengukuran kepuasan pengguna terhadap layanan ini belum dilakukan secara terintegrasi dengan sistem *ticketing* yang sudah ada. Pengumpulan umpan balik masih dilakukan dengan metode tradisional, sehingga sulit digunakan secara konsisten untuk mengevaluasi kualitas layanan.

Selain pengumpulan umpan balik, organisasi juga memerlukan kemampuan analisis agar data survei tidak berhenti pada tahap merekap nilai. Salah satu pendekatan yang dapat digunakan adalah analisis berbasis pengelompokan (*cohort analysis*) untuk mengelompokkan dan memantau ringkasan informasi kepuasan pengguna pada periode tertentu, misalnya berdasarkan jenis layanan atau waktu penggunaan [7]. Dengan menghubungkan data tiket *helpdesk* dan hasil survei kepuasan, UPA TIK berpotensi memperoleh ringkasan terkait penggunaan ulang, partisipasi dalam pengisian survei, serta dinamika nilai kepuasan dari waktu ke waktu, yang disajikan dalam bentuk pelaporan deskriptif sebagai bahan *monitoring*.

Berbagai solusi survei digital, seperti *Google Form*, masih memiliki keterbatasan dalam hal fleksibilitas dan analisis data. Beberapa platform survei *online* juga sudah menyediakan logika perhitungan hasil survei, salah satunya *SurveyMonkey*. Namun, fitur yang disediakan sangat terbatas untuk pengguna versi gratis [8]. Banyak penelitian yang telah menghasilkan aplikasi survei dalam bentuk *website*, serta mampu menampilkan statistik kepuasan pengguna [3], [4]. Selain itu, sudah ada penelitian yang membahas hasil kepuasan layanan *helpdesk*, sebagai indikator kesuksesan layanan digital [5], [6]. Namun, belum ada penelitian yang secara eksplisit menghasilkan fitur kepuasan yang terintegrasi langsung dengan layanan *helpdesk*, serta menyediakan pengelompokan (*cohort*) dan penyajian statistik untuk menggambarkan variasi karakteristik pengguna.

Berdasarkan uraian masalah tersebut, penelitian ini bertujuan untuk merancang dan membangun aplikasi *mobile* yang terintegrasi dengan sistem *helpdesk* di UPA TIK

Universitas Lampung. Aplikasi ini dirancang untuk memudahkan pengumpulan umpan balik terstruktur karena survei dikaitkan dengan layanan/tiket tertentu, memungkinkan pemilihan jenis penilaian yang fleksibel dengan mekanisme kuesioner yang dapat dikustomisasi, serta dapat mengetahui karakteristik pengguna pada kelompok atau periode tertentu. Sistem juga memanfaatkan fitur *push notification* sebagai pengingat otomatis kepada pengguna untuk meningkatkan partisipasi dalam pengisian survei dan memastikan umpan balik terkumpul tepat waktu setelah tiket ditangani.

Dalam implementasinya, sistem dikembangkan menggunakan *flutter* untuk aplikasi Android dan web admin untuk performa yang tinggi dan efisiensi CPU dalam pengembangan aplikasi *multi-platform* [9], *Go (Golang)* untuk *backend* API karena *throughput* dan performa yang tinggi [10], serta *PostgreSQL* sebagai basis data relasional. Pengembangan sistem menggunakan metodologi *Rapid and Participatory Application Development* (RAPD), yaitu kombinasi *Rapid Application Development* (RAD) untuk pengembangan yang cepat dan *Participatory Design* (PD) untuk memastikan keterlibatan pengguna dalam setiap tahapan desain dan pengambilan keputusan, sehingga aplikasi yang dihasilkan benar-benar sesuai dengan kebutuhan pengguna [11], [12], [13].

Untuk memastikan aplikasi yang dibangun berfungsi dan dapat diterima pengguna, dilakukan pengujian bertahap, meliputi pengujian *white-box* dengan teknik unit testing pada level komponen, pengujian fungsional dengan metode *black-box* pada level fitur [14], serta dilakukan pengujian *User Acceptance Test* (UAT) yang melibatkan pengguna akhir sebagai penilai penerimaan sistem [15] dengan strategi prioritas pengujian menggunakan pendekatan *risk-based testing* agar fokus pengujian lebih dahulu diberikan pada fungsi yang paling kritis terhadap sistem. Aplikasi ini diharapkan dapat membantu UPA TIK dalam mengumpulkan dan menganalisis data kepuasan pelanggan, sehingga dapat menjadi dasar pengambilan keputusan dalam upaya meningkatkan kualitas layanan.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang dan mengembangkan aplikasi survei kepuasan pengguna berbasis *mobile* yang terintegrasi dengan sistem *helpdesk* UPA TIK Universitas Lampung?
2. Bagaimana merancang fitur kustomisasi kuesioner pada sistem kepuasan yang memungkinkan pengguna memberikan penilaian terhadap tiket layanan *helpdesk* yang telah selesai ditangani?
3. Bagaimana merancang fitur rekapitulasi hasil survei secara menyeluruh dan mengimplementasikan analisis berbasis pengelompokan (*cohort analysis*) untuk menyajikan data survei berdasarkan kelompok dan periode waktu tertentu?

## 1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Merancang dan mengembangkan aplikasi survei kepuasan pengguna berbasis *mobile* yang terintegrasi dengan sistem *helpdesk* UPA TIK Universitas Lampung.
2. Merancang fitur kustomisasi kuesioner pada sistem yang memungkinkan pengguna memberikan penilaian terhadap tiket layanan *helpdesk* yang telah selesai ditangani.
3. Merancang fitur rekapitulasi hasil survei secara menyeluruh serta mengimplementasikan analisis berbasis pengelompokan (*cohort analysis*) untuk menyajikan data survei berdasarkan kelompok dan periode waktu tertentu.

## 1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini, di antaranya:

1. Bagi UPA TIK Universitas Lampung: membantu memperoleh umpan balik pengguna secara terstruktur dan terdokumentasi sebagai dasar perbaikan layanan secara bertahap.
2. Bagi civitas academica Universitas Lampung: mempermudah penyampaian penilaian dan masukan terhadap layanan melalui media yang dapat diakses kapan saja dan di mana saja.
3. Bagi institusi (Universitas Lampung): mendukung penyelenggaraan pelayanan publik yang terukur melalui tersedianya data umpan balik dan rekapitulasi hasil survei sebagai bahan evaluasi kualitas layanan.
4. Bagi pengembangan sistem dan penelitian selanjutnya: menjadi referensi implementasi modul survei kepuasan yang terintegrasi dengan sistem layanan yang sudah berjalan.

## 1.5 Batasan Penelitian

Penelitian ini memiliki batasan, yaitu:

1. Penelitian berfokus pada pengembangan modul survei kepuasan pengguna yang terintegrasi dengan sistem *helpdesk* UPA TIK Unila; proses *ticketing* pada *helpdesk* bukan objek pengembangan utama; karena keterbatasan akses ke *codebase*/DB/API pada sistem berjalan, penelitian menggunakan replikasi minimal fitur tiket untuk menyediakan konteks layanan dan kebutuhan pengujian.
2. Kategori layanan mengikuti kategori pada sistem *helpdesk* yang sudah ada, sehingga tidak terdapat fitur penambahan, edit, atau penghapusan kategori pada sistem survei ini.
3. Formulir umpan balik ditampilkan kepada pengguna pada tiket dengan status selesai; untuk kebutuhan pengujian modul survei pada penelitian ini, semua tiket yang dibuat dianggap berstatus selesai.

4. Pengembangan aplikasi difokuskan pada platform *mobile* untuk pengguna dan *website* untuk admin, dengan *Flutter* sebagai *framework* utama.
5. Pengujian yang dibahas berfokus pada kebutuhan fungsional; pengujian unit testing tidak dilakukan pada seluruh fungsi logika sistem, melainkan diprioritaskan pada fungsi murni dengan risiko *high* dan *medium*; fungsi lain secara keseluruhan diuji melalui *blackbox testing* dan UAT.

## 1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini bertujuan untuk mempermudah dalam menyusun, mengelola, merangkai, dan menulis data hasil penelitian. Adapun sistematika penulisan adalah sebagai berikut:

- I. **PENDAHULUAN:** Bagian ini memuat latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan.
- II. **TINJAUAN PUSTAKA:** Bagian ini memuat dasar-dasar teori yang digunakan dalam penelitian ini, termasuk dasar metodologi serta berisi kajian penelitian yang berkaitan.
- III. **METODE PENELITIAN:** Pada bagian ini menjelaskan metodologi yang digunakan pada penelitian ini. Pada bagian ini juga memuat waktu dan tempat penelitian, serta alat dan bahan yang digunakan.
- IV. **HASIL DAN PEMBAHASAN:** Bagian ini menjelaskan hasil yang diperoleh pada penelitian, meliputi hasil pengembangan dan hasil pengujian, serta memuat pembahasan pada penelitian.
- V. **KESIMPULAN:** Bagian ini memuat kesimpulan dari hasil penelitian yang telah dilakukan, serta memuat saran-saran untuk penelitian lanjutan.

**DAFTAR PUSTAKA:** Pada bagian ini memuat seluruh referensi yang digunakan dalam proses penelitian.

**LAMPIRAN:** Bagian ini memuat lampiran-lampiran penelitian.

## II. TINJAUAN PUSTAKA

### 2.1 Kepuasan Pengguna

Kepuasan pengguna didefinisikan sebagai ukuran perasaan pengguna terhadap interaksi dengan produk, merek, atau layanan. Kepuasan pengguna adalah cerminan dari sejauh mana harapan pengguna telah terpenuhi atau terlampaui. Dalam perspektif bisnis, mengukur kepuasan dianggap penting. Jika pelanggan merasa puas terhadap suatu produk, mereka memiliki kecenderungan untuk terus menggunakan produk tersebut, bahkan memengaruhi orang lain untuk menggunakan produk serupa. Hal ini tentu memberikan keuntungan bagi perusahaan, karena secara tidak langsung mendapatkan promosi secara gratis melalui pelanggan mereka. Dengan mengukur kepuasan pengguna, perusahaan dapat mengidentifikasi kekurangan mereka untuk terus meningkatkan kualitas. Dengan begitu, perusahaan dapat menjalin hubungan jangka panjang dengan pelanggannya [16].

Dalam literatur sistem informasi, kepuasan pengguna digunakan sebagai indikator keberhasilan sistem. *User Information Satisfaction* (UIS) didefinisikan sebagai sejauh mana pengguna merasa yakin bahwa sistem informasi yang tersedia memenuhi kebutuhan (*requirements*) informasi mereka. UIS mengukur cara pengguna memandang sistem yang ada berdasarkan persepsi mereka. Bukan semata-mata kualitas teknis sistem, bahkan sistem yang secara teknis sudah baik, bisa dianggap buruk bila dipersepsikan buruk oleh pengguna [17].

## 2.2 Survei Kepuasan

Survei kepuasan pada dasarnya merupakan kegiatan pengumpulan data yang menggunakan kuesioner sebagai instrumen utama. Brace mendefinisikan kuesioner sebagai wawancara terstruktur yang telah disiapkan dan bersifat tetap (*fixed*), artinya setiap responden akan menerima pertanyaan yang sama dengan cara yang sama, sehingga data yang terkumpul lebih konsisten untuk dianalisis. Pertanyaan-pertanyaan dapat mencakup beragam subjek, seperti preferensi pribadi, keyakinan, pengalaman, perilaku, atau informasi demografis. Tujuan penulisan kuesioner adalah untuk memperoleh jawaban atau data yang paling akurat dari responden. Dalam bisnis dan pemasaran, survei membantu organisasi dalam mengukur kepuasan pelanggan, mengumpulkan umpan balik mengenai produk atau layanan, yang dapat digunakan untuk mengidentifikasi area yang perlu ditingkatkan. Dalam praktiknya, survei dapat dilakukan melalui berbagai media, seperti wawancara, tatap muka/telepon, maupun isian mandiri seperti kuesioner kertas atau berbasis *online*.

Untuk mengukur kepuasan, survei biasanya memakai *rating scales*. Skala penilaian memungkinkan pemangku kepentingan untuk mengukur intensitas, atau kepuasan responden terhadap suatu objek, layanan, atau atribut. Pemilihan jumlah kategori skala, misalnya 4, 5, atau 7 poin perlu disesuaikan dengan kebutuhan diskriminasi jawaban, kemampuan responden, dan media yang digunakan. Misalnya jika media yang digunakan adalah melalui panggilan telepon, maka skala lebih dari 5 poin akan cenderung sulit untuk diimplementasikan. Selain itu, perlu dipertimbangkan apakah akan menyediakan opsi netral atau “tidak tahu”, karna keputusan ini memengaruhi interpretasi.

Bahasa yang digunakan pada kuesioner juga perlu diperhatikan agar tidak terjadi ambiguitas, pertanyaan ganda, atau pertanyaan yang bersifat mengarahkan. Brace juga menekankan pentingnya *piloting* (uji coba) kuesioner sebelum survei utama diluncurkan. Uji coba sebaiknya melibatkan responden dengan karakteristik yang beragam, agar masalah pemahaman pertanyaan dapat terlihat sejak awal [18].

*Rating scale* seperti yang disebutkan hanyalah salah satu dari beberapa format jawaban yang tersedia dalam perancangan kuesioner. Secara lebih luas, Kumar mengklasifikasikan format jawaban dalam penelitian bisnis ke dalam dua kategori besar, yaitu *open-ended* (tidak terstruktur) dan *closed-ended* (terstruktur). Pada format *open-ended*, responden bebas merumuskan jawaban mereka sendiri tanpa batasan opsi yang ditentukan sebelumnya, sehingga mampu menangkap perspektif dan nuansa yang lebih kaya. Sebaliknya, format *closed-ended* memberikan pilihan jawaban yang telah ditetapkan peneliti, yang kemudian dapat diklasifikasikan lebih lanjut berdasarkan skala pengukurannya, meliputi pertanyaan dikotomis (*dichotomous*) pada skala nominal yang hanya menyediakan dua pilihan, seperti Ya atau Tidak, pertanyaan pilihan ganda (*multiple choice*) dengan lebih dari dua opsi tidak berurutan, skala penilaian bertingkat (*rating scale*), termasuk skala *Likert* pada skala ordinal, serta pertanyaan peringkat (*rank order*) yang meminta responden mengurutkan preferensi. Pada pertanyaan tertutup berbasis skala nominal, dua bentuk respons yang banyak digunakan dalam riset bisnis adalah pertanyaan dikotomis dan pilihan ganda [19].

Sementara itu, Dolnicar dan Grün menunjukkan bahwa dalam praktik riset survei, format ordinal multi-kategori seperti skala *Likert* merupakan yang paling dominan digunakan. Penelitian ini juga menunjukkan bahwa tidak ada satu format jawaban yang optimal untuk semua konstruk, sehingga pemilihan format jawaban perlu mempertimbangkan kesesuaian antara jenis konstruk dan bentuk respons yang diharapkan. Dengan demikian, penggunaan variasi format jawaban dalam satu instrumen kuesioner dapat diterima, selama pemilihannya didasarkan pada kesesuaian dengan sifat konstruk yang diukur [20].

### 2.3 Helpdesk

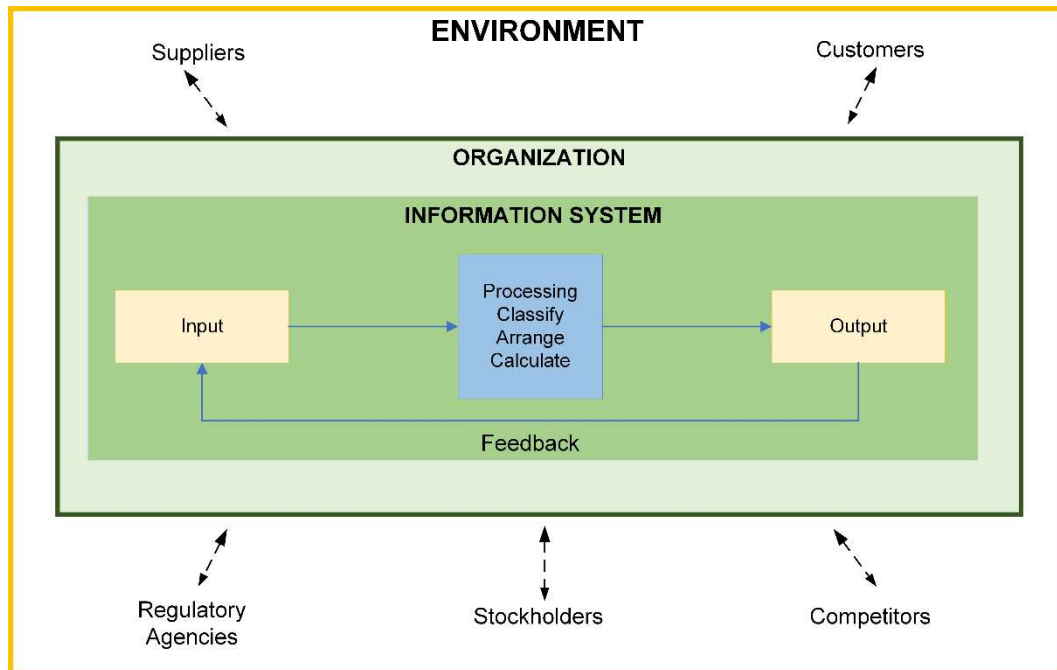
Dalam praktik layanan Teknologi Informasi, *helpdesk* sering digunakan sebagai istilah untuk layanan yang menangani permintaan bantuan dari pengguna. Dalam kerangka ITIL (*Information Technology Infrastructure Library*) 4, istilah ini disebut sebagai *service desk*. ITIL menempatkan *service desk practice* sebagai praktik yang berfokus pada penangkapan kebutuhan pengguna terkait penyelesaian *incident* dan *service request*, serta berperan sebagai pintu utama antara penyedia layanan dan seluruh pengguna. Insiden merupakan gangguan yang tidak direncanakan terhadap suatu layanan atau dapat juga dianggap sebagai penurunan kualitas layanan. Sementara *service request* didefinisikan sebagai permintaan bantuan dari pengguna yang telah ditentukan dan disepakati sebagai bagian dari layanan yang diberikan (contoh: permintaan mengatur ulang kata sandi, permintaan perbaikan jaringan dan sebagainya). ITIL 4 juga menjelaskan keterkaitan antara *service desk* dengan dua praktik operasional utama, yaitu:

1. *Incident management*, yang bertujuan untuk meminimalkan dampak negatif insiden dengan memulihkan operasi layanan menjadi normal kembali sesegera mungkin.
2. *Service request management*, yang bertujuan untuk menangani permintaan layanan yang telah didefinisikan sebelumnya secara efektif dan ramah pengguna.

Secara konseptual, *helpdesk/service desk* dapat didefinisikan sebagai fungsi layanan yang mengintegrasikan penerimaan laporan, pencatatan, klasifikasi, prioritas, eskalasi, pemantauan status, serta komunikasi kepada pengguna hingga isu terselesaikan. Dalam implementasi modern, mekanisme tersebut umumnya difasilitasi oleh *ticketing system*, sehingga laporan pengguna tidak hanya terselesaikan, tapi juga dapat ditelusuri (*traceable*) untuk kebutuhan evaluasi mutu layanan dan perbaikan berkelanjutan [21].

## 2.4 Sistem Informasi

Secara teknis, sistem informasi adalah kumpulan komponen yang saling terhubung, berupa perpaduan perangkat keras, perangkat lunak, dan jaringan telekomunikasi yang berfungsi untuk menghimpun, mengolah, menyimpan, dan menyalurkan informasi guna mendukung proses pengambilan keputusan serta pengendalian dalam suatu organisasi.



Gambar 2.1 Fungsi Sistem Informasi [22].

Gambar 2.1 menggambarkan tiga aktivitas utama dalam sistem informasi, yaitu *input*, proses, dan *output*, yang bersama-sama menghasilkan informasi yang dibutuhkan organisasi untuk mendukung pengambilan keputusan, mengendalikan operasi, menganalisis berbagai permasalahan, serta mengembangkan produk atau layanan baru. Selain itu, sistem informasi juga menegaskan perbedaan antara data sebagai fakta mentah yang belum diolah dengan informasi sebagai data yang telah diproses sehingga memiliki makna dan kegunaan bagi penggunanya [22].

## 2.5 Cohort Analysis

*Cohort analysis* didefinisikan sebagai bentuk *behavioral analytics* yang mengelompokkan pengguna berdasarkan karakteristik yang sama, kemudian melacak perilakunya secara periodik untuk menemukan pola, seperti retensi, frekuensi penggunaan, dan lain-lain. Salah satu tantangan analisis data pada sistem adalah jumlahnya yang sangat besar dan beragam. Dengan menggunakan *cohort analysis*, data dikelompokkan berdasarkan karakteristiknya, sehingga pola dapat terlihat jelas, lalu analisis per kelompok memberi pemahaman lebih spesifik atas perilaku kelompok tersebut. Ketika perilaku dipahami, dapat menjadi masukan bagi pengambilan keputusan dan pengembangan suatu produk atau layanan.

Cohorts											
											Returning Visitors %
COHORT ▲	WEEK 0	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10
Feb 13–19, 2023	100%	98.4%	98.69%	89.67%	98.98%	98.01%	98.36%	98.7%	92.94%	98.73%	98.76%
Feb 20–26, 2023	100%	97.71%	88.61%	100%	99.7%	100%	100%	95.14%	100%	100%	100%
Feb 27–Mar 5, 2023	100%	47.85%	91.94%	88.71%	89.25%	93.55%	73.66%	94.09%	94.09%	95.16%	94.62%
Mar 6–12, 2023	100%	59.09%	68.18%	63.64%	50%	59.09%	54.55%	63.64%	63.64%	54.55%	40.91%
Mar 13–19, 2023	100%	60.77%	81.62%	95.46%	57.72%	97.89%	86.16%	81.7%	79.7%	53.34%	75.05%
Mar 20–26, 2023	100%	69.13%	61.74%	29.53%	86.58%	67.79%	39.6%	34.9%	28.86%	35.57%	34.9%
Mar 27–Apr 2, 2023	100%	52.5%	21.07%	48.21%	20.36%	14.64%	12.5%	11.07%	10.71%	10.71%	1.43%
Apr 3–9, 2023	100%	42.86%	50%	20.41%	5.1%	5.1%	5.1%	5.1%	5.1%	3.06%	2.04%
Apr 10–16, 2023	100%	33.33%	20.83%	8.33%	8.33%	4.17%	4.17%	4.17%	0%	0%	0%
Apr 17–23, 2023	100%	6.9%	3.45%	3.45%	3.45%	3.45%	3.45%	3.45%	0%	3.45%	3.45%

Gambar 2.2 *Cohort Analysis* Retensi Pengguna.

(sumber: <https://matomo.org/blog/2024/01/cohort-analysis-marketing/>)

Gambar 2.2 merupakan contoh *heatmap* retensi penggunaan berulang berdasarkan *cohort* tertentu. Pembentukan *cohort* dapat menggunakan atribut tertentu yang relevan dengan tujuan analisis. Contoh atribut yang sering digunakan adalah gender, tanggal penggunaan/pembelian pertama dan terakhir, tingkat penggunaan, serta atribut lain yang dapat membedakan perilaku antar kelompok [23].

## 2.6 Teknologi Pengembangan Sistem

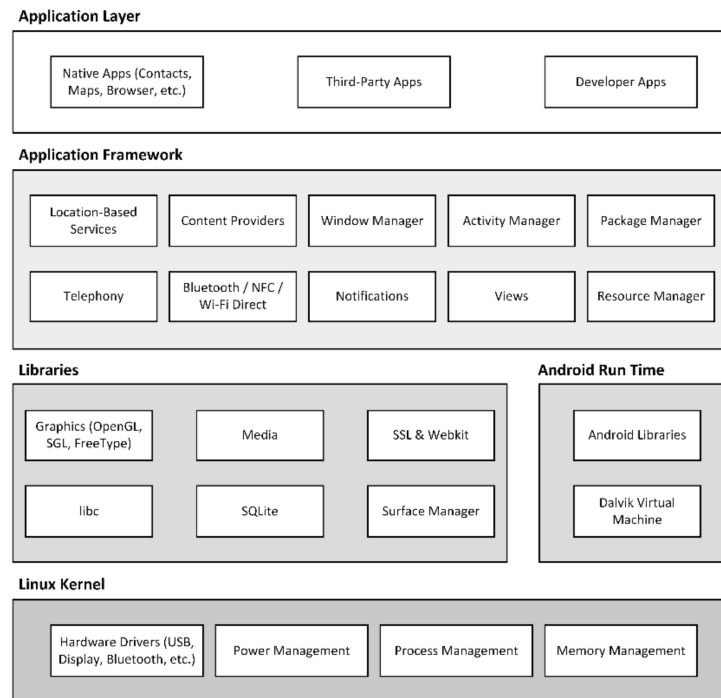
### 2.6.1 *Go*

*Go*, atau *Golang*, merupakan bahasa pemrograman *open source* yang dikembangkan oleh *Robert Griesemer*, *Rob Pike*, dan *Ken Thompson* di *Google*. Bahasa ini bersifat terkompilasi, memiliki tipe data statis, serta dirancang untuk kesederhanaan, efisiensi, dan keterbacaan yang tinggi. *Go* telah mendapatkan popularitas karena keserbagunaannya, dukungan konkurensi bawaannya, dan perpustakaan standarnya yang lengkap. Tujuan *Go* adalah untuk membuat bahasa pemrograman yang mudah dipelajari dan digunakan, tetapi juga kuat dan efisien untuk membangun sistem perangkat lunak yang kompleks. Beberapa kelebihan *Golang*, di antaranya:

1. Kesederhanaan dan kemudahan penggunaan.
2. Dukungan konkurensi bawaan melalui *goroutines* dan *channels*.
3. *Library* standar yang lengkap.
4. Performa dan efisiensi yang cepat.
5. Dukungan yang berkembang dari komunitas sumber terbuka [24].

### 2.6.2 *Android*

*Android* adalah sistem operasi dengan basis *linux* mencakup sistem operasi, *middleware*, dan aplikasi yang dijalankan pada perangkat *mobile*. Telepon pintar yang menggunakan sistem operasi ini lebih populer dan diterima masyarakat dibandingkan dengan jenis operasi lainnya. *Android* merupakan sistem operasi dengan sumber daya terbuka (*open source*) yang membuat para *developer* tidak memerlukan biaya pengembangan yang besar. Karena sifat *Android* yang *open source* inilah yang menyebabkan pengembang aplikasi berbasis *Android* bermunculan.



Gambar 2.3 Arsitektur *Android* [25].

Gambar 2.3 menampilkan susunan arsitektur sistem operasi Android yang terdiri atas lima komponen utama yang dikelompokkan ke dalam empat lapisan:

1. *Kernel Linux*

Pada lapisan paling dasar terdapat *Kernel Linux* yang berfungsi sebagai inti sistem, berisi berbagai *driver* perangkat keras serta menangani pengelolaan proses, memori, keamanan, jaringan, dan konsumsi daya perangkat.

2. *Libraries*

Di atasnya terdapat lapisan *Libraries* yang memuat kumpulan pustaka C/C++ yang menyediakan beragam fungsi dasar Android, misalnya *SQLite* yang menyediakan fasilitas basis data untuk keperluan penyimpanan data aplikasi.

3. *Android runtime*

*Android runtime* menyediakan himpunan pustaka inti dan lingkungan eksekusi sehingga pengembang dapat menulis aplikasi Android menggunakan bahasa pemrograman Java. Di dalam *Android runtime* terdapat *Dalvik Virtual Machine*, yaitu mesin virtual yang membuat setiap aplikasi berjalan dalam proses terpisah dan dikompilasi ke dalam format khusus *Dalvik*, serta dioptimalkan untuk perangkat bergerak yang memiliki keterbatasan baterai, memori, dan kemampuan CPU.

#### 4. *Application Framework*

Menyediakan kelas yang digunakan dalam pengembangan aplikasi *Android*.

#### 5. *Application Layer*

Semua aplikasi, baik yang sudah terinstal secara bawaan maupun yang dikembangkan oleh pihak ketiga, berada di lapisan ini [25].

### 2.6.3 *Dart*

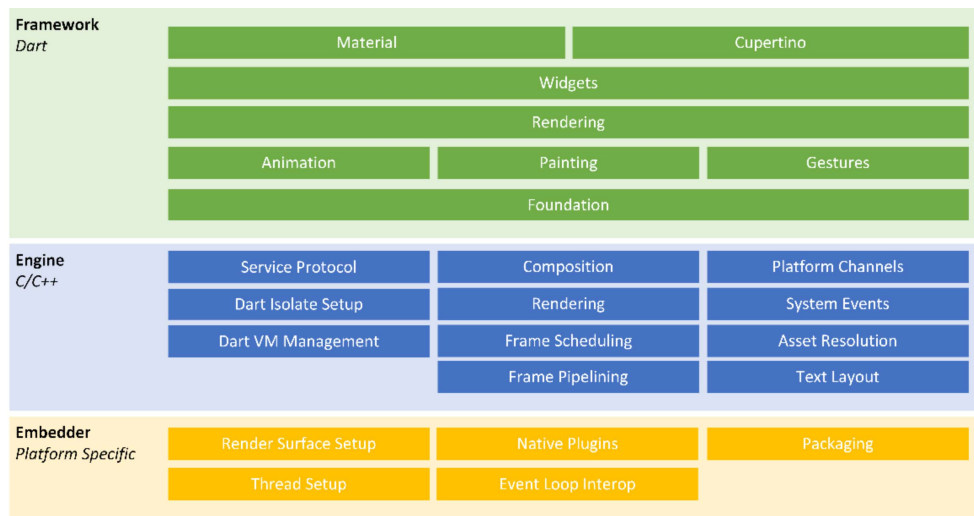
*Dart* adalah bahasa pemrograman *multiplatform* dan memiliki karakteristik *open source* yang awalnya dibuat untuk menggantikan dan melanjutkan *JavaScript*. Dengan demikian, *dart* menggunakan sebagian besar karakteristik dari standar *JavaScript* (ES7), seperti kata kunci “*async*” dan “*wait*”. Ini pertama kali dirancang oleh *Google* pada tahun 2011 sebelum disetujui oleh ECMA sebagai standar. *Dart* adalah bahasa yang dioptimalkan untuk klien yang memungkinkan Anda membuat aplikasi dengan cepat pada platform apa pun. Sebagai bahasa pemrograman modern, *Dart* dioptimalkan untuk menangani skenario ini di tingkat memori dengan bantuan “*Generational Garbage Collection*”. *Dart* dirancang agar mudah dipelajari dan digunakan, menjadikannya pilihan yang baik untuk pemula. *Dart* merupakan bahasa pemrograman yang berlandaskan paradigma pemrograman berorientasi objek berbasis kelas, yang dimaksudkan untuk membangun aplikasi berkualitas tinggi untuk perangkat seluler dan *web*.

*Dart* memiliki beberapa fitur yang membuatnya menjadi bahasa yang menarik untuk digunakan, termasuk:

1. Pengetikan opsional: *Dart* memungkinkan untuk menggunakan pengetikan statis atau dinamis.
2. Kemudahan penggunaan: *Dart* dirancang agar mudah dipelajari dan digunakan.
3. Keserbagunaan: *Dart* dapat digunakan untuk membangun berbagai aplikasi.
4. Kinerja tinggi: *Dart* dirancang untuk membangun aplikasi yang berkinerja tinggi.
5. Dukungan alat yang hebat: *Dart* memiliki sejumlah alat yang hebat yang tersedia, termasuk *compiler*, *debugger*, dan IDE [26].

### 2.6.4 Flutter

*Flutter* adalah kerangka kerja *multiplatform* yang dirancang oleh Google dengan tujuan untuk membuat pengembangan perangkat lunak *multiplatform* menjadi lebih mudah dengan basis kode yang lebih sederhana. *Flutter* berbeda dengan bahasa pengembangan lain yang memisahkan basis kode untuk UI, karena *flutter* menerapkan satu basis kode yang mencakup UI dan *logic*. *Flutter* menggunakan bahasa pemrograman *Dart* dan memiliki arsitektur berlapis yang memberikan kinerja tinggi dan pengalaman pengguna yang lancar. Salah satu keunggulan utama *Flutter* adalah kemampuan *hot reload*-nya, yang memungkinkan developer untuk melihat perubahan kode secara *real-time* tanpa harus mengompilasi ulang aplikasi. Dengan *Flutter*, developer dapat menggunakan basis kode tunggal untuk mengembangkan aplikasi yang dapat dijalankan pada berbagai platform, termasuk *iOS*, *Android*, *Windows*, *macOS*, *Web*, dan *Linux*.



Gambar 2.4 Arsitektur *Flutter* [27].

Gambar 2.4 merupakan arsitektur *flutter* yang terdiri dari tiga lapisan pengembangan, yaitu *embedder*, *engine*, dan *framework layer*. *Flutter* menyediakan beragam *widget* UI yang dapat disesuaikan dan telah dioptimalkan untuk berbagai platform. Selain itu, *Flutter* memiliki mesin *render* sendiri yang memungkinkan aplikasi *Flutter* berjalan dengan lancar di berbagai perangkat. *Rendering Flutter* berjalan pada 60 *frame per second* (*fps*) dan 120 *fps* (untuk perangkat yang memiliki kemampuan 120Hz). Semakin besar *fps*, semakin lancar aplikasi yang dijalankan [26], [27].

### 2.6.5 PostgreSQL

*PostgreSQL* merupakan sebuah sistem manajemen basis data berbasis objek-relasional (*object-relational database management system/ORDBMS*) yang dirilis pada 1989. *PostgreSQL* awalnya merupakan proyek POSTGRES yang dipimpin oleh Michael Stonebraker, profesor di *University of California, Berkeley*. *PostgreSQL* tidak hanya basis data *SQL-compliant*, tetapi menawarkan fitur-fitur modern, seperti:

1. Memungkinkan untuk *queries* yang kompleks.
2. *Foreign keys* untuk menghubungkan data dalam dua tabel.
3. *Triggers* yang secara otomatis dipicu pada saat proses *input* dan pengecekan, konfirmasi, mengubah, menghapus, atau menggunakan data referensi secara elektif.
4. Tampilan yang dapat diperbaharui
5. Konsep transaksi yang lengkap
6. *Multi-version concurrency control* (MVCC) untuk eksekusi akses basis data yang efisien.

*PostgreSQL* adalah salah satu basis data paling populer, selama beberapa dekade, *postgres* telah berkembang menjadi produk tingkat *enterprise* yang kini menjadi alternatif nyata untuk basis data komersial. Oleh karena itu *postgres* menyediakan dukungan untuk server siaga aktif (*hot standby*), *point-in-time recovery*, dan berbagai jenis replikasi (*synchronous, asynchronous, cascade*). Dari sisi keamanan, *postgres* mendukung koneksi SSL dan menyediakan metode autentikasi. *PostgreSQL* juga menyediakan dukungan penuh untuk properti ACID (*Atomicity, Consistency, Isolation, Durability*) dan isolasi transaksi yang efisien berdasarkan *multi-version concurrency control* (MVCC). *PostgreSQL* memanfaatkan arsitektur CPU *multi-core* modern sehingga kinerjanya meningkat secara linear, seiring bertambahnya jumlah *core*. *PostgreSQL* dapat memaralelkan *queries* dalam beberapa perintah sekaligus, seperti pembuatan indeks dan *vacuuming* [28], [29].

### 2.6.6 *Firebase Cloud Messaging (FCM)*

*Firebase* merupakan platform *Backend-as-a-Service* (BaaS) milik Google yang menyediakan seperangkat alat dan layanan untuk membantu *developer* dalam membangun dan mengelola aplikasi *web* maupun *mobile*. Dengan menggunakan platform ini, pengembang tidak perlu lagi melakukan konfigurasi dan pengelolaan server secara manual, sehingga bisa fokus pada pengembangan fitur aplikasi. *Firebase* menawarkan berbagai layanan yang mencakup kebutuhan pengembang dari awal hingga akhir, mulai dari autentikasi pengguna (*Firebase Authentication*), penyimpanan data secara *realtime* (*Firebase Realtime Database*) dan berbasis dokumen (*Cloud Firestore*), eksekusi kode *backend* tanpa kelola server (*Cloud Functions*), penyajian aplikasi *web* (*Firebase Hosting*), pengiriman notifikasi *push* (*Firebase Cloud Messaging/FCM*), analitik penggunaan aplikasi (*Firebase Analytics*), hingga penyimpanan berkas seperti gambar dan video (*Firebase Cloud Storage*).

*Firebase Cloud Messaging (FCM)* adalah salah satu produk *Firebase* untuk berkomunikasi dengan pengguna. FCM memungkinkan pengiriman pesan secara andal dan efisien daya ke aplikasi klien di berbagai platform, baik *Android*, *iOS*, dan *web*. Dengan FCM, aplikasi dapat mengirim notifikasi bahwa data baru perlu disinkronkan, serta mengirim *notifications message* untuk mendorong keterlibatan pengguna. Implementasi FCM mencakup dua komponen utama, yaitu lingkungan yang terpercaya untuk membangun dan mengirim pesan, serta aplikasi klien untuk menerima pesan.

Setiap aplikasi perlu mengambil *token* pada perangkat FCM untuk dapat menerima pesan. *Token* bersifat unik untuk setiap instalasi aplikasi dan harus dikirimkan ke server *backend* agar server dapat menargetkan pengiriman pesan ke perangkat yang spesifik. FCM mendukung dua jenis pesan, yaitu *notification messages* (umumnya ditangani sistem/SDK untuk ditampilkan) dan *data messages* (diproses oleh kode logika aplikasi untuk menentukan tindakan) [30].

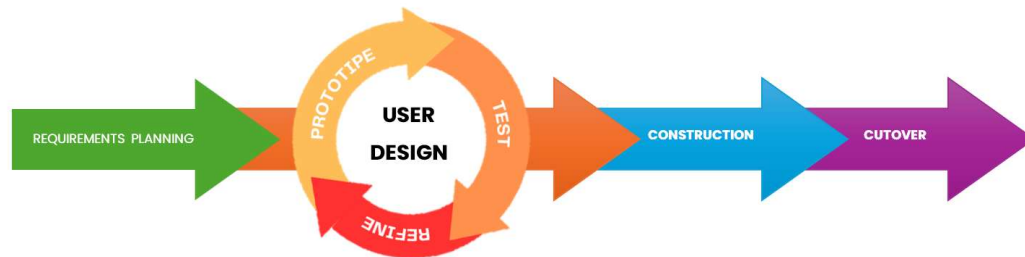
## 2.7 Metode Pengembangan Sistem

### 2.7.1 *Rapid Application Development (RAD)*

RAD (*Rapid Application Development*) menjadi alternatif dari metode tradisional dengan siklus pengembangan lebih cepat dengan kualitas lebih baik yang dirancang pada tahun 1970-an dan pertama kali diperkenalkan pada tahun 1991. Saat ini RAD berperan sebagai salah satu metode yang paling kuat untuk mengelola proyek kecil dan menengah. Pada metode tradisional *waterfall*, pengguna dipaksa untuk menandatangani persyaratan setelah menyelesaikan setiap spesifikasi sebelum pengembangan dapat dilanjutkan dan dilakukan pengembangan, yang menyebabkan pengguna tidak dapat melihat hasil apa pun sebelum sistem selesai diimplementasikan dan proses pengembangan dapat memakan waktu yang sangat lama. RAD dianggap sebagai solusi yang lebih cepat dibandingkan dengan metode *waterfall*. Meskipun pengembangan dilakukan dengan cepat, bukan berarti RAD memiliki makna "*quick and dirty*". Karena fokusnya pada pembuatan prototipe dan keterlibatan pengguna dalam fase analisis, desain, pembuatan, dan pengujian dari siklus hidup pengembangan, RAD menghasilkan kualitas *software* yang lebih baik. Setiap tahapan dikompresi menjadi serangkaian siklus pengembangan yang pendek dan berulang. Model RAD digunakan untuk proyek dengan struktur yang jelas dan mudah dipahami. Dasar-dasar dari metodologi RAD meliputi:

1. Menggabungkan teknik-teknik terbaik dan menentukan urutan tugas yang paling efektif.
2. Menggunakan prototipe evolusioner yang nantinya diubah menjadi produk akhir.
3. Menggunakan *workshop*, alih-alih wawancara untuk mengumpulkan persyaratan dan meninjau sistem.
4. Memilih alat bantu CASE (*Computer-Aided Software Engineering*) untuk membantu proses pengembangan aplikasi.
5. Menerapkan pengembangan terjadwal yang memungkinkan tim bekerja dengan cepat dan mengimplementasikan penyempurnaan untuk rilis berikutnya.
6. Memberikan panduan untuk keberhasilan dan menjelaskan hal-hal yang harus dihindari.

Keterlibatan pengguna pada metode RAD memastikan persyaratan bisnis dan harapan pengguna dipahami dengan jelas. Siklus hidup RAD dirancang untuk membangun sebuah sistem yang mencukupi kebutuhan pengguna dan melewati empat tahapan utama.



Gambar 2.5 Siklus Hidup *Rapid Application Development* (RAD) [31].

Gambar 2.5 merupakan Siklus hidup RAD yang terdiri dari empat tahapan utama, yaitu:

1. *Requirements Planning*

Pada tahap ini mendefinisikan proses bisnis dan area subjek data yang akan didukung sistem, serta menentukan ruang lingkup sistem.

2. *User Design*

Untuk memodelkan data dan proses sistem, serta membuat prototipe dari komponen yang penting, melalui *workshop*.

3. *Construction*

Pada tahap ini, menyelesaikan pengembangan sistem, membuat kebutuhan pengguna, dan membuat rencana implementasi.

4. *Cutover*

Pada tahap ini, pengujian dan pelatihan pengguna akhir, serta implementasi sistem aplikasi, dilakukan [31], [32].

### 2.7.2 *Participatory Design (PD)*

*Participatory Design (PD)* adalah metode desain yang melibatkan pengguna secara aktif dan kolaboratif dalam proses perancangan dan pengembangan suatu produk atau sistem. Dengan melibatkan mereka secara langsung dalam proses pengambilan keputusan desain, tujuannya adalah untuk menghasilkan solusi yang lebih relevan, mudah digunakan, dan memiliki nilai bagi pengguna akhir. Metode ini bermula dari gerakan demokrasi di tempat kerja Skandinavia antara tahun 1970-an dan juga 1980-an, pada saat itu para pekerja terlibat pada desain sistem teknologi untuk meningkatkan kondisi kerja mereka. Saat ini, PD telah berkembang dalam berbagai bidang, termasuk desain interaksi manusia dengan komputer (*Human-Computer Interaction / HCI*) [33]. Adapun prinsip utama PD, yaitu:

1. Mengutamakan kolaborasi aktif antara desainer dan pengguna. Pengguna berfungsi sebagai mitra dalam proses desain dan sebagai sumber informasi [11].
2. Memberdayakan pengguna dengan memberi mereka pengaruh dalam proses desain. Hal ini memungkinkan pengguna untuk menyuarakan kebutuhan, keinginan, dan ide-ide.
3. PD mendorong pembelajaran timbal balik antara desainer dan pengguna. Desainer memahami konteks pengguna, sementara pengguna mengetahui kendala dan potensi desain [33].
4. PD selalu menempatkan pengguna sebagai fokus utama. Semua keputusan desain harus didasarkan pada pemahaman yang mendalam tentang kebutuhan dan karakteristik pengguna [12].

#### 2.7.2.1 *Generasi Pertama Participatory Design (PD)*

*Participatory Design (PD)* telah mengalami perkembangan yang cukup signifikan sejak kemunculannya pada tahun 1970-an. Pendekatan ini dimulai dengan melibatkan warga sipil secara aktif dalam merencanakan lingkungan hidup. Pendekatan tersebut dengan cepat meluas ke pekerja industri dengan tujuan untuk meningkatkan demokrasi di tempat kerja dan meningkatkan otonomi pekerja, keahlian, dan variasi tugas yang biasa disebut dengan "*Collective Systems Design*". Pengguna ikut serta langsung pada desain melalui partisipasi mereka dalam tim desain. Desain sistem kolektif menggunakan alat bantu yang mudah dipelajari dan

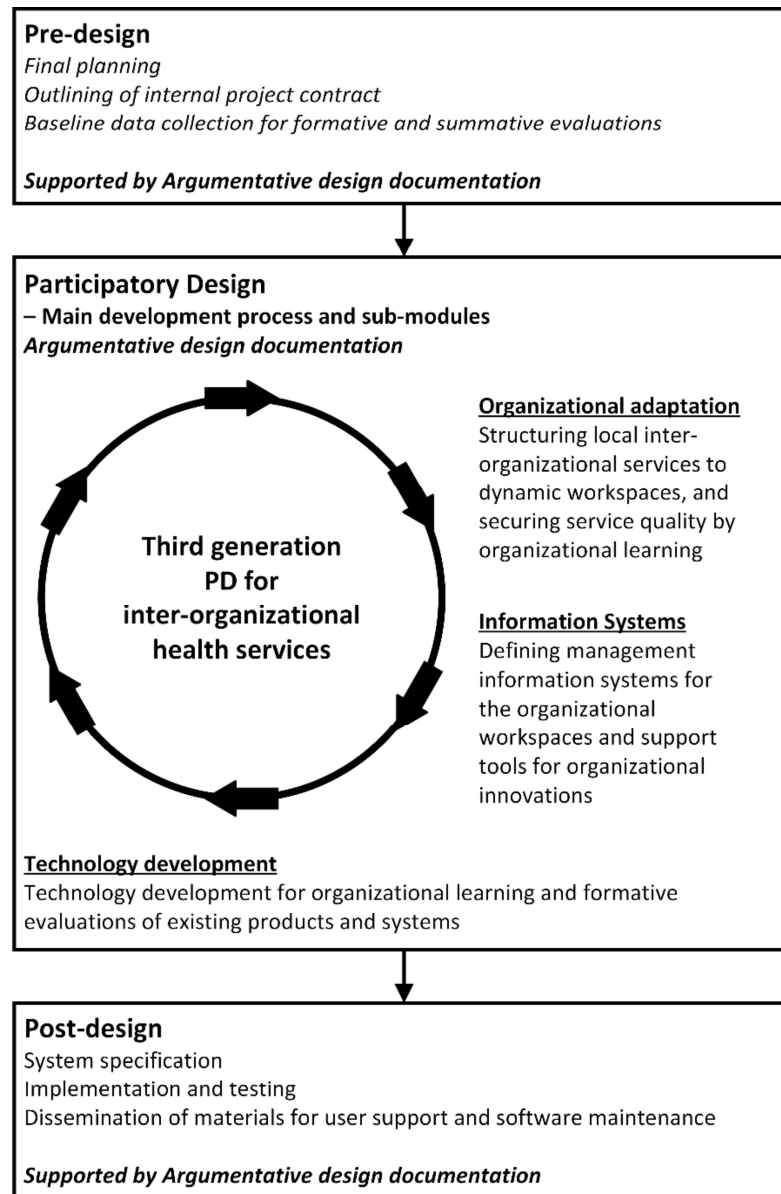
tidak terlalu menuntut pengetahuan pengguna, seperti *mock-up*, *future workshops*, dan PICTIVE (*low-tech* material untuk desain *graphical user interfaces* (GUI)). Namun, tantangan dalam menjalankan pendekatan ini juga menjadi perhatian, misalnya kurangnya formalisasi yang mengakibatkan peningkatan kompleksitas implementasi secara keseluruhan. masalah lain juga ditemukan pada pengguna yang kurang berpartisipasi dan cenderung ingin menyerahkan masalah ini kepada para ahli dan hanya fokus pada kebutuhan informasi [34].

#### **2.7.2.2 Generasi Kedua *Participatory Design* (PD)**

Pada tahun 1980-an serta 1990-an awal, istilah “*Computer Support Cooperative Work*” (CSCW) muncul mengacu pada peningkatan solusi teknologi informasi. Pendekatan ini dikembangkan secara paralel dengan *Collective Systems Design*, sehingga istilah *participatory design* mulai digunakan. Pada generasi ini menghasilkan sistem yang secara umum lebih baik karena dirancang bersama pengguna secara langsung. Namun, generasi PD ini belum banyak digunakan pada pengembangan dengan skala besar. Pendekatan ini hanya menghasilkan aplikasi TI berskala kecil [34].

#### **2.7.2.3 Generasi Ketiga *Participatory Design* (PD)**

PD generasi ketiga menuntut adaptasi terhadap tren organisasi yang umum, misalnya pada proyek berskala besar, kolaborasi dan jaringan antar organisasi, serta menjangkau secara luas orang-orang yang bekerja di organisasi. Hal ini menghasilkan kelompok pengguna yang semakin berkembang dan heterogen. Gagasan pemangku kepentingan dengan pengguna akhir menjadi sorotan karna partisipasi pengguna aktif masih dapat diperdebatkan dari perspektif demokrasi. Namun, konflik organisasi dapat diselesaikan pada tahap awal, sehingga sistem memiliki peluang lebih tinggi untuk diterima dan digunakan.



Gambar 2.6 Generasi Ketiga *Participatory Design* [34].

Gambar 2.6 merupakan tahapan pada generasi ketiga PD. Masalah dan kesulitan pada generasi sebelumnya menghasilkan revisi substansial terhadap metode desain. Kerangka kerja diperbaharui didasarkan pada tiga modul dengan modul kedua memiliki tiga sub-modul. Modul pertama berisi kegiatan sebelum PD dimulai. Meskipun hanya modul kedua yang berisi proses inti PD, namun pada modul pertama dan ketiga juga mencakup elemen partisipasi pengguna, misalnya dengan memberikan umpan balik tentang garis besar rencana proyek, serta memberikan evaluasi terhadap prototipe yang dikembangkan. Sementara itu pada modul kedua

didasarkan pada desain sistem kolektif. Setiap pertemuan digunakan untuk membahas permasalahan yang dapat mencakup beberapa praktik desain. Aturan untuk argumentasi desain ditetapkan pada pertemuan pertama, misalnya bagaimana menentukan prioritas permasalahan dan bagaimana menghasilkan keputusan desain [34].

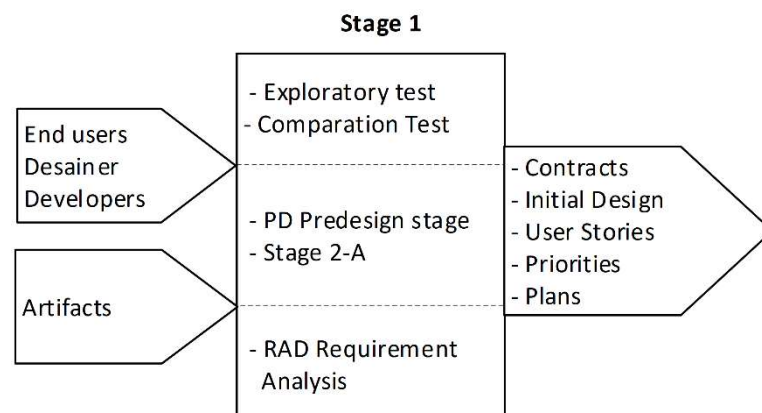
### 2.7.3 *Rapid and Participatory Application Development (RAPD)*

*Rapid and Participatory Application Development (RAPD)* merupakan kerangka kerja pengembangan aplikasi yang menggabungkan *Rapid Application Development (RAD)* untuk mempercepat proses pengembangan dengan *Participatory Design (PD)*, lalu menambahkan *usability testing* pada setiap fase yang digunakan sebagai mekanisme verifikasi dan validasi desain. Pada RAPD, RAD dan PD diintegrasikan dengan cara “menumpang-tindihkan” tahapan desain dengan partisipasi pengguna untuk mempercepat produksi. Tujuan integrasi RAD dan PD di antaranya:

1. Memanfaatkan RAD untuk pengembangan dan penyerahan sistem yang cepat.
2. Menggunakan PD untuk kolaborasi pengguna dalam proses desain dan menangkap semua detail dalam konteks kerja.
3. Menambahkan *usability testing* untuk verifikasi dan validasi desain sistem di semua tahapan.

Secara sederhana, RAPD dijelaskan dalam tiga tahap besar, yaitu:

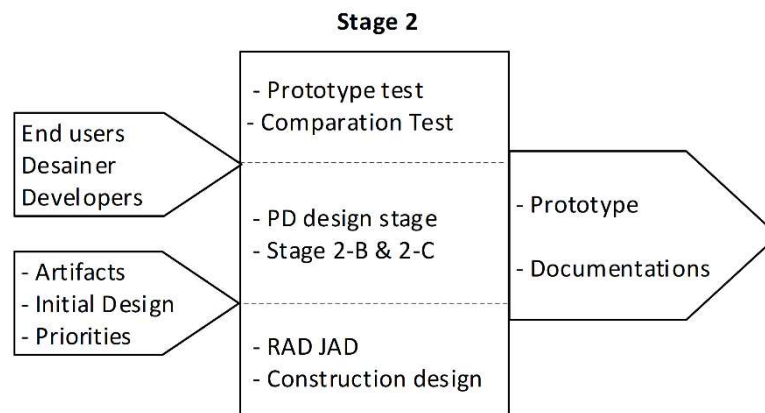
1. *RAPD Stage 1*



Gambar 2.7 *Stage 1 RAPD* [12].

Gambar 2.7 merupakan tahap integrasi dari tahap perencanaan pada RAD dengan tahap pra-desain PD dan PD *stage 2-A*. Desainer dan pengguna menentukan tujuan, rencana, prioritas, dan jadwal, serta membentuk tim desainer dan perwakilan pengguna untuk mengumpulkan kebutuhan, mengidentifikasi semua entitas, dan menggambar semua diagram dan interaksi antara fungsi dan data. Desainer dan pengguna melakukan pertemuan untuk melakukan *exploratory usability testing* untuk memastikan kebutuhan telah sesuai.

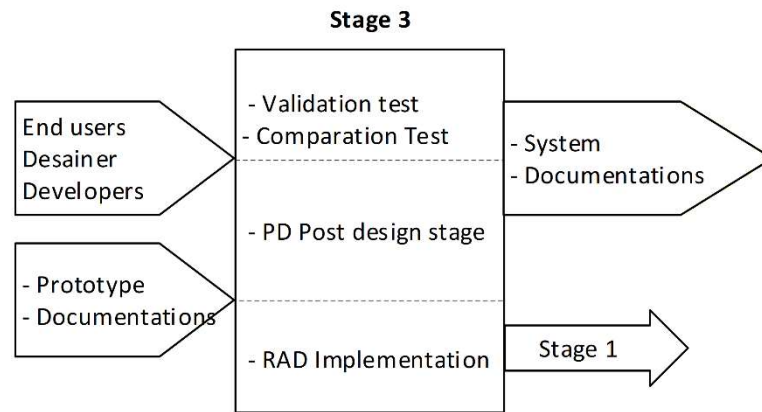
## 2. RAPD *Stage 2*



Gambar 2.8 *Stage 2* RAPD [12].

Gambar 2.8 merupakan tahap integrasi dari RAD tahap 2 (JAD) dan tahap 3 (*construction stage*) dengan PD 2-B dan 2-C. Tahap ini memungkinkan pengembang untuk membangun prototipe awal berdasarkan *output* yang disepakati pada tahap 1. Prototipe awal diunggah ke *cloud* untuk dilakukan uji coba oleh perwakilan pengguna akhir. Pengujian didasarkan pada *test case* yang telah disusun untuk mempermudah pengguna akhir dalam melakukan uji coba fitur sistem dan dapat memberikan umpan balik. Dokumentasi yang berkaitan dengan perubahan *requirements*, *user stories*, dan desain dikumpulkan untuk pemeliharaan dan modifikasi ke depan.

### 3. RAPD Stage 3



Gambar 2.9 Stage 2 RAPD [12].

Gambar 2.9 merupakan tahap integrasi dari tahap implementasi pada RAD dengan tahap pasca-desain PD untuk mengimplementasikan prototipe menjadi sistem final dan menguji sistem akhir. Pada tahap ini semua dokumen ditinjau ulang untuk memastikan semua kebutuhan telah dipertimbangkan. Setiap tahap juga berisi *comparison testing* untuk memungkinkan mencari desain dan solusi alternatif apabila terdapat kebutuhan yang perlu diubah.

Dalam RAPD, pengujian *usability* tidak diposisikan sebagai pengujian akhir saja, tetapi disisipkan di setiap tahapan (*exploratory test*, *prototyping test*, *validation test*, dan *comparison test*) agar rancangan, prototipe, sampai produk akhir terus divalidasi dari sisi kemudahan dan kesesuaian dengan kebutuhan [11], [12], [13].


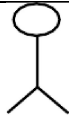

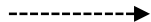


### 2.8 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) dikembangkan sebagai bahasa standar dalam pemodelan sistem, tidak terbatas pada perangkat lunak saja. UML memungkinkan analis, arsitek, dan pengembang perangkat lunak untuk mendefinisikan, memvisualisasikan, membangun, dan mendokumentasikan arsitektur suatu sistem secara terstruktur. Setiap diagram UML memungkinkan pengguna dan pengembang melihat sistem dari berbagai sudut pandang. Diagram UML dibuat dalam alat bantu pemodelan visual, meliputi [35]:

### 2.8.1 Use Case Diagram

Sebuah *use case diagram* merupakan representasi visual yang menggambarkan bagaimana sebuah sistem berinteraksi dengan satu atau beberapa aktor yang terlibat. Diagram ini juga memaparkan perilaku fungsionalitas sistem, sub sistem, kelas, atau komponen dari sudut pandang pengguna. Aktor tidak hanya merujuk pada manusia, tetapi juga mencakup proses atau entitas eksternal lainnya yang berinteraksi dengan sistem, sub sistem, atau kelas tertentu. Adapun komponen *use case*, dideskripsikan pada Tabel 2.1.



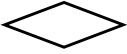

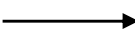

Tabel 2.1 Komponen *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Use case</i>	Menggambarkan fungsi sebuah sistem.
	<i>Actor</i>	Mewakili orang-orang yang berinteraksi dengan sistem.
	<i>Association</i>	Mewakili bagaimana aktor berinteraksi dalam kasus tertentu.
	<i>Extend</i>	<i>Use case</i> yang ditambahkan dapat berdiri sendiri untuk menunjukkan perilaku tambahan. Sebuah <i>use case</i> ditambahkan dengan arah panah.
	<i>Generalization</i>	Menggambarkan hubungan antara <i>use case</i> umum dan <i>use case</i> yang lebih khusus yang mewarisi dan menambahkan fitur. Contoh umum ditunjukkan dengan arah panah.
	<i>Include</i>	Merepresentasikan perilaku baru ke dalam <i>use case</i> yang secara eksplisit, menggunakan <i>use case</i> dasar untuk menjalankan fungsinya. Tanda panah mengarah kepada <i>use case</i> yang ditambahkan.

### 2.8.2 Activity Diagram

Aktivitas diagram adalah grafik yang menunjukkan aliran data operasional dan kontrol melalui proses komputasi. *Node* aktivitas merepresentasikan eksekusi dari sebuah prosedur dalam sistem. Eksekusi langkah-langkah dapat dilakukan secara berurutan atau bersamaan (paralel). Simbol dalam *activity diagram* dideskripsikan pada Tabel 2.2.

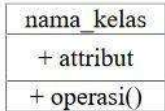
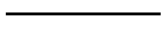
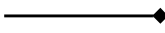

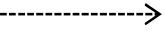
Tabel 2.2 Komponen *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point/Initial State</i>	Merepresentasikan status awal suatu aktivitas.
	<i>Activity</i>	Menggambarkan tindakan yang dilakukan oleh sistem ( <i>node</i> aktivitas).
	<i>Decision</i>	Merepresentasikan terdapat lebih dari satu pilihan aktivitas.
	Penggabungan/ <i>Join</i>	Merepresentasikan penggabungan lebih dari satu aktivitas digabung menjadi satu.
	<i>Transition</i>	Merepresentasikan petunjuk aktivitas, baik aktivitas sebelum ataupun sesudahnya.
	<i>Final State/End State</i>	Merepresentasikan akhir dari aktivitas.

### 2.8.3 *Class Diagram*

Dengan menggunakan elemen seperti kelas, paket, dan objek, *class diagram* dapat memodelkan struktur dan isi kelas. Ini merepresentasikan sistem berorientasi objek yang juga menampilkan hubungan pewarisan, asosiasi, generalisasi, dan berbagai macam ketergantungan. Komponen *class diagram* dideskripsikan pada Tabel 2.3.







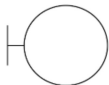

Tabel 2.3 Komponen *Class Diagram*

Simbol	Nama	Deskripsi
	Kelas	Mewakili kelas dalam struktur sistem, setiap kelas memiliki nama, atribut, dan operasi atau metode.
	Asosiasi	Merepresentasikan koneksi antar kelas.
	Agregasi	Merepresentasikan hubungan antar kelas dengan salah satu kelas merupakan bagian dari kelas lain dan dapat berdiri masing-masing.
	Generalisasi	Merepresentasikan hubungan antar kelas dengan pengertian generalisasi-spesialisasi (umum-khusus), digunakan untuk pewarisan dan deklarasi tipe polimorfik.
	Ketergantungan	Merepresentasikan hubungan antara dua elemen kelas.

### 2.8.4 Sequence Diagram

*Sequence diagram* menggambarkan urutan waktu dari semua objek yang terlibat dalam interaksi. Diagram ini memiliki dua sumbu, yakni sumbu vertikal yang merepresentasikan alur waktu yang bergerak dari atas ke bawah, serta sumbu horizontal yang merepresentasikan objek-objek yang berpartisipasi dalam interaksi tersebut. Secara umum, *sequence diagram* hanya menunjukkan urutan pesan dan bukan interval waktu yang tepat. Sebuah pesan ditampilkan sebagai panah dari *life line* satu objek ke objek lainnya. Panah-panah tersebut disusun berdasarkan urutan waktu pada diagram. Komponen *sequence diagram* dideskripsikan pada Tabel 2.4.

Tabel 2.4 Komponen *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Mewakili entitas di luar sistem; ini dapat termasuk orang, proses, dan sistem lain.
	Objek	Menyatakan item yang berhubungan dengan pesan.
	<i>Life Line</i>	Menyatakan aktivitas dari suatu objek.
	Waktu Aktif	Menunjukkan objek yang berinteraksi dan dalam keadaan aktif.
	<i>Message</i>	Menggambarkan komunikasi yang terjadi antar objek dan tindakan yang akan dilakukan
	<i>Entity Class</i>	Merepresentasikan penyimpanan atau data sistem.
	<i>Boundary Class</i>	Merupakan objek yang berinteraksi dengan aktor sistem.
	<i>Control Class</i>	Merupakan perantara antara <i>entity</i> dan <i>boundary</i> . <i>Control</i> mengatur pelaksanaan perintah yang berasal dari <i>boundary</i> dengan <i>entity</i> .

## 2.9 Pengujian Perangkat Lunak

### 2.9.1 *White-box Testing*

Pengujian *white-box* adalah pengujian perangkat lunak yang berfokus pada pemeriksaan detail alur prosedural yang ada di dalam sistem. Tujuan utamanya adalah memverifikasi kebenaran *statement*, *code paths*, *conditions*, *loops*, dan *data flow*. Pengujian ini dilakukan dengan menelusuri jalur logika pada perangkat lunak sekaligus menguji keterkaitan antar komponen melalui serangkaian kondisi atau perulangan tertentu. Sekilas, pengujian ini dianggap akan menghasilkan program yang 100% benar. Pengembang hanya mendefinisikan semua jalur logis, mengembangkan kasus uji dan menjalankannya, kemudian mengevaluasi hasilnya. Sayangnya, jumlah kemungkinan jalur logis bisa sangat besar yang tidak memungkinkan melakukan pengujian secara menyeluruh. Oleh karena itu, pengembang dapat memilih sejumlah jalur logis yang penting dan terbatas serta memeriksa validitas struktur data penting. Meskipun demikian, semakin tinggi cakupan pengujian logika, semakin kecil kemungkinan akan ditemukan *bug* oleh penguji lain. Semakin awal *bug* ditemukan, maka akan semakin murah biaya perbaikannya [14].

### 2.9.2 *Unit Testing*

*Unit testing* adalah jenis pengujian yang berfokus pada bagian terkecil dari sebuah desain perangkat lunak, yakni komponen atau modul individual. Pengujian ini diarahkan pada pemeriksaan logika pemrosesan internal serta struktur data yang berada dalam cakupan komponen tersebut. Karena ruang lingkungannya kecil, kompleksitas dan jenis kesalahan yang ditemukan juga relatif terbatas. Pengujian ini dapat dijalankan secara bersamaan terhadap beberapa komponen sekaligus. Hal-hal yang diperiksa pada *unit testing*, di antaranya:

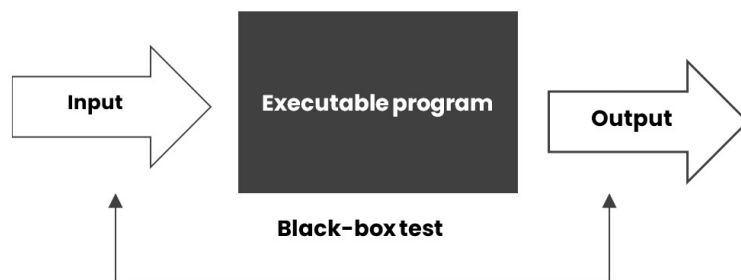
1. Antarmuka modul: memastikan data mengalir dengan benar. Pengujian ini dilakukan pada tahap awal, karena jika terdapat kesalahan, tes lain menjadi tidak relevan.
2. Struktur data lokal: memastikan integritas data selama eksekusi program.

3. Jalur kontrol penting/jalur independen: menjalankan jalur-jalur eksekusi agar setiap pernyataan pada modul dieksekusi minimal satu kali.
4. Kondisi batas (*boundary*): menguji nilai minimal/maksimal, indeks *array*, batas *loop*, serta nilai tepat di batas bawah atau batas atas.
5. *Error handling*: kesalahan yang diuji mencakup pesan *error* yang tidak jelas, *error* yang dicatat tidak sesuai, intervensi sistem sebelum *handler* berjalan, salah menangani *exception*, atau info *error* yang tidak membantu menemukan penyebab.

Desain *test* dapat dilakukan sebelum atau sesudah proses *coding*, dengan memastikan tiap kasus uji memiliki hasil sesuai dengan yang diharapkan. Sayangnya, banyak komponen tidak dapat diuji secara memadai dengan perangkat yang sederhana. Dalam kasus seperti ini, pengujian lengkap dapat ditunda sampai *integration testing* [14].

### 2.9.3 *Black-box Testing*

Pengujian *black-box* merupakan pengujian yang tidak memperhatikan mekanisme internal karena pengembang tidak perlu mengetahui kode program dari sistem yang sedang dibuat. Kode dianggap sebagai kotak hitam dan penguji hanya mengetahui apa yang *diinputkan* ke dalam kotak, kemudian kotak akan mengirimkan *output*. Pengujian didasarkan pada kebutuhan fungsional sistem dengan perspektif pengguna akhir dengan tujuan memastikan sistem aplikasi yang dikembangkan dapat dijalankan sesuai dengan persyaratan. *Blackbox testing* diilustrasikan pada Gambar 2.10.



Gambar 2.10 Metode *Black-box Testing* [36].

Setiap kebutuhan yang tidak sesuai akan dengan mudah diidentifikasi dan ditangani. *Black-box testing* bukan merupakan pengganti dari *white-box*, melainkan bersifat komplementer, karena pendekatan ini mampu mengungkap kategori kesalahan yang berbeda dari yang ditemukan melalui metode *white-box*. *Black-box testing* bertujuan untuk mendeteksi kesalahan dalam beberapa kategori berikut: (1) fungsi yang tidak berjalan sebagaimana mestinya atau bahkan tidak tersedia sama sekali, (2) kesalahan pada antarmuka sistem, (3) kesalahan dalam struktur data maupun saat mengakses basis data eksternal, (4) ketidaksesuaian pada perilaku atau performa sistem, serta (5) kesalahan yang muncul pada saat proses inisialisasi maupun penghentian sistem [14].

Dalam perancangan kasus uji *black-box*, terdapat beberapa teknik yang dapat digunakan, di antaranya *equivalence partitioning* (pembagian domain masukan ke dalam kelas valid dan tidak valid), *boundary value analysis* (pengujian pada nilai batas domain masukan), *cause-effect graphing* (pemetaan hubungan logis antara kondisi masukan dan keluaran), *error guessing* (menebak area kesalahan berdasarkan pengalaman penguji), serta *decision table testing* [36].

Salah satu teknik yang umum digunakan untuk menguji fungsi dengan banyak kombinasi kondisi logis adalah *decision table testing*. *Decision table* memodelkan kombinasi kondisi masukan dan tindakan yang dihasilkan sistem secara sistematis dalam bentuk tabel, di mana setiap kolom merepresentasikan satu aturan keputusan (*rule*) yang menggambarkan kombinasi kondisi tertentu beserta keluaran yang diharapkan. Dengan cara ini, seluruh kombinasi kondisi yang relevan dapat tercakup secara menyeluruh dan terdokumentasi tanpa ada kemungkinan yang terlewat.

Struktur *decision table* terdiri dari dua bagian utama, yaitu bagian kondisi (*condition*) dan bagian aksi (*action*). Bagian kondisi memuat daftar kondisi yang dievaluasi beserta nilai kemungkinannya, sedangkan bagian aksi memuat tindakan yang dihasilkan sistem untuk setiap kombinasi kondisi. Setiap kolom merepresentasikan satu aturan keputusan (*rule*) yang diberi label R1, R2, R3, dan

seterusnya. Kondisi ditandai dengan notasi Y apabila terpenuhi, N apabila tidak terpenuhi, dan - apabila tidak relevan terhadap aturan tersebut, sedangkan aksi yang dihasilkan ditandai dengan Y apabila terjadi dan N apabila tidak terjadi. Tabel 2.5 adalah struktur *decision table* berdasarkan format yang digunakan oleh Pressman [14].

Tabel 2.5 Contoh Struktur *Decision Table* [14]

Kondisi / Aksi	R1	R2	R3	R4
<b>Kondisi</b>				
C1: <i>Username</i> terdaftar	Y	N	Y	Y
C2: <i>Password</i> benar	Y	-	N	Y
C3: Akun aktif	Y	-	-	N
<b>Aksi</b>				
A1: <i>Login</i> sukses, <i>token</i> diterbitkan	Y	N	N	N
A2: Pesan " <i>Username</i> tidak ditemukan"	N	Y	N	N
A3: Pesan " <i>Password</i> salah"	N	N	Y	N

Keterangan:

Y = kondisi terpenuhi/aksi terjadi;

N = kondisi tidak terpenuhi/aksi tidak terjadi;

- = tidak relevan.

Perancangan kasus uji *black-box testing* juga dapat dilakukan menggunakan pendekatan berbasis skenario (*scenario-based testing*). Pada pendekatan ini, kasus uji dirancang berdasarkan skenario alur pengguna secara *end-to-end* melalui antarmuka aplikasi, dengan tujuan memverifikasi bahwa sistem berfungsi sebagaimana mestinya dari sudut pandang pengguna akhir tanpa memperhatikan struktur internal kode. Setiap kasus uji didokumentasikan dalam bentuk tabel yang memuat skenario pengujian, hasil yang diharapkan, hasil aktual, dan kesimpulan pengujian [14].

#### 2.9.4 Risk-Based Testing

*Risk-Based Testing* (RBT) adalah pendekatan pengujian sistem yang objek pengujiannya dipilih, diprioritaskan, dan dikelola berdasarkan analisis dan pengendalian risiko. *International Software Testing Qualifications Board* (ISTQB) menjelaskan bahwa RBT merupakan bagian dari manajemen risiko yang memungkinkan organisasi meningkatkan kemungkinan tercapainya tujuan (mengacu pada ISO 31000), memperbaiki kualitas produk, serta meningkatkan kepercayaan pemangku kepentingan [37]. Amland mendefinisikan ide dasar RBT untuk memfokuskan pengujian dan mengalokasikan lebih banyak waktu pada fungsi-fungsi *critical*. Amland menjelaskan bahwa proyek-proyek TI sangat jarang terselesaikan tepat waktu atau dalam anggaran yang telah ditetapkan. Fase-fase awal proyek sering mengalami keterlambatan, sehingga pada saat pengujian, hanya memiliki sisa waktu yang terbatas, serta anggaran telah habis karena pembengkakan biaya pengembangan. Oleh karena itu, diperlukan suatu metodologi untuk membantu memprioritaskan apa saja yang perlu diuji dan menentukan seberapa banyak kasus uji agar tetap menghasilkan perangkat lunak yang berkualitas dengan tenggat waktu yang tersisa [38]. Sejalan dengan itu, ISTQB menegaskan bahwa pengujian menyeluruh (*exhaustive testing*) adalah hal yang tidak mungkin dilakukan, dan sebagai gantinya, teknik-teknik pengujian, prioritas kasus uji, serta *risk-based testing* harus digunakan untuk memfokuskan upaya pengujian [37].

ISTQB mendefinisikan *risk* sebagai potensi kejadian, bahaya, ancaman, atau situasi yang apabila terjadi akan menimbulkan dampak negatif. Suatu risiko didasari oleh dua faktor utama, yaitu: *risk likelihood* (probabilitas terjadinya risiko) dan *risk impact* (konsekuensi atau kerugian akibat terjadinya risiko tersebut). Kedua faktor ini membentuk *risk level*, yang merupakan ukuran pentingnya suatu risiko. Semakin tinggi *risk level*, semakin penting untuk ditangani, dan pengujian akan diprioritaskan. ISTQB membedakan dua jenis risiko yang relevan untuk pengujian, yaitu *project risks* (bisa berdampak pada jadwal/biaya/ruang lingkup proyek) dan *product risks* (berkaitan dengan kualitas produk, misalnya fungsionalitas salah/hilang, kesalahan perhitungan, *runtime error*, arsitektur buruk, performa tidak memadai, UX buruk, dan kerentanan keamanan). Jika terjadi, *product risks* dapat

menyebabkan ketidakpuasan pengguna, hilangnya reputasi/kepercayaan, biaya pemeliharaan tinggi, hingga konsekuensi yang sangat serius pada sistem. ISTQB menjelaskan bahwa penilaian risiko dapat menggunakan pendekatan kuantitatif dengan perhitungan sebagai berikut [37]:

$$\text{Risk Level} = \text{Risk Likelihood} \times \text{Risk Impact} \quad (2.1)$$

Dari perspektif pengujian, tujuan analisis risiko produk (*product risk analysis*) adalah memberikan kesadaran akan risiko produk untuk memfokuskan upaya pengujian sehingga meminimalkan tingkat risiko produk. Analisis risiko produk idealnya dimulai sedini mungkin dalam SDLC dan terdiri atas identifikasi risiko dan penilaian risiko. Hasil dari analisis risiko digunakan untuk menentukan ruang lingkup pengujian, *test level* dan *test type* yang diperlukan, teknik pengujian dan target *coverage*, estimasi *effort*, prioritas eksekusi pengujian agar *defect critical* ditemukan lebih awal, serta menentukan aktivitas tambahan di luar testing jika diperlukan untuk mengurangi risiko [37], [38].

### 2.9.5 User Acceptance Testing (UAT)

*User Acceptance Testing* (UAT) adalah tahap akhir dalam proses pengujian sebelum sistem diterima untuk penggunaan operasional. Sistem diuji dengan data pengguna akhir, bukan menggunakan data simulasi. Pengujian ini dapat mengungkap kesalahan dan kelalaian dalam pendefinisian persyaratan sistem, karena data yang digunakan adalah data nyata dengan prosedur yang berbeda dari data uji. UAT juga dapat mengungkap permasalahan persyaratan sistem jika sistem tidak benar-benar memenuhi kebutuhan pengguna atau kinerja sistem yang tidak sesuai. Terdapat enam tahapan dalam proses UAT, di antaranya [15]:

#### 1. Menentukan *acceptance criteria*

Idealnya, tahap ini dilakukan pada proses awal sebelum kontrak untuk pengembangan sistem ditandatangani. Kriteria penerimaan harus sesuai dengan kontrak awal yang telah disepakati, meskipun dalam praktiknya bisa saja terdapat perubahan persyaratan di tengah proses pengembangan.

2. Merencanakan pengujian

Tahap ini melibatkan penentuan sumber daya, waktu, anggaran untuk pengujian, dan penetapan jadwal. Tahap ini juga harus membahas cakupan persyaratan yang dibutuhkan dan urutan pengujian sistem.

3. Menurunkan pengujian *acceptance*

Pengujian harus dirancang untuk memeriksa apakah suatu sistem dapat diterima atau tidak, meliputi pengujian fungsional dan non-fungsional.

4. Menjalankan pengujian

Pengujian idealnya dilakukan pada *environment* yang sebenarnya, yang menjadi tempat sistem dijalankan. Namun, mungkin ini akan mengganggu dan tidak praktis, sehingga perlu menyiapkan lingkungan pengujian khusus. Beberapa pelatihan pengguna mungkin diperlukan sebelum menjalankan pengujian.

5. Negosiasi hasil pengujian

Dalam pengujian, tidak mungkin 100% kasus uji dapat lulus tanpa ada kendala. Jika terdapat masalah, pengembangan dan pelanggan harus bernegosiasi guna memutuskan apakah sistem yang telah dikembangkan sudah cukup baik untuk digunakan.

6. Penolakan/penerimaan sistem

Tahap ini melibatkan pengguna dan pengembang dalam memutuskan penerimaan sistem. Jika sistem tidak cukup baik untuk digunakan, maka pengembang perlu memperbaiki masalah yang teridentifikasi dan mengulangi fase pengujian *acceptance*.

Dalam pelaksanaannya, UAT memerlukan instrumen yang dapat mengukur penerimaan pengguna secara sistematis dan terstandar. Kuesioner merupakan instrumen yang umum digunakan dalam pengujian UAT karena memungkinkan pengumpulan data penilaian dari pengguna akhir secara terstruktur dan dapat dikuantifikasi. Melalui kuesioner, pengguna diminta memberikan penilaian terhadap aspek-aspek sistem seperti fungsionalitas, antarmuka, kemudahan penggunaan, dan kesesuaian dengan kebutuhan, yang kemudian dianalisis sebagai dasar keputusan penerimaan sistem [39].

Penilaian UAT menggunakan kuesioner dapat menggunakan skala *Likert* 1–5, dengan kategori untuk setiap butir pertanyaan bernilai 1 = Sangat Tidak Setuju, 2 = Tidak Setuju, 3 = Kurang Setuju, 4 = Setuju, dan 5 = Sangat Setuju. Skala *Likert* digunakan karena kemampuannya dalam mengonversi penilaian kualitatif responden menjadi data kuantitatif yang dapat diolah secara matematis, sekaligus menghasilkan interpretasi kualitatif yang mudah dipahami oleh para pemangku kepentingan. Proses perhitungan skor UAT dilakukan melalui tiga tahap yang saling berkaitan. Pertama, seluruh nilai jawaban responden dijumlahkan untuk memperoleh total skor aktual. Kedua, dihitung skor maksimal (ideal) sebagai batas tertinggi yang mungkin dicapai jika seluruh responden memilih nilai tertinggi. Ketiga, kedua nilai tersebut dibandingkan untuk menghasilkan persentase skor UAT yang merepresentasikan tingkat penerimaan pengguna terhadap sistem secara keseluruhan [40]. Ketiga tahap perhitungan skor ini dideskripsikan pada Tabel 2.6.

Tabel 2.6 Tahap Perhitungan Skor UAT

Tahap	Komponen	Rumus / Keterangan
1.	<b>Total Skor Aktual</b>	$\Sigma(\text{Jumlah Responden} \times \text{Skor Likert})$ (2.2)
2.	<b>Skor Maksimal (Ideal)</b>	$\text{Skor Max} = n \times \text{Skala Likert Tertinggi}$ (2.3) Keterangan: $n = \text{Jumlah responden}$
3.	<b>Persentase Skor UAT</b>	$\text{Index (\%)} = \frac{\text{Total Skor Aktual}}{\text{Skor Maksimal}} \times 100\%$ (2.4)

Hasil persentase skor UAT kemudian diinterpretasikan menggunakan rentang kategori penerimaan yang dijelaskan pada Tabel 2.7. Kategori ini berfungsi sebagai acuan objektif dalam menentukan apakah sistem yang dikembangkan telah memenuhi standar kelayakan untuk digunakan oleh pengguna akhir. Semakin tinggi persentase yang diperoleh, semakin kuat indikasi bahwa sistem telah berhasil memenuhi kebutuhan dan ekspektasi pengguna.

Tabel 2.7 Rentang Persentase Penerimaan

<b>Rentang Persentase</b>	<b>Kategori</b>	<b>Keterangan</b>
<b>80%-100%</b>	<b>Sangat Baik/Sangat Layak</b>	Sistem diterima dan layak digunakan tanpa perbaikan berarti
<b>60%-79,99%</b>	<b>Baik/Layak</b>	Sistem dapat diterima dengan perbaikan minor
<b>40%-59,99%</b>	<b>Cukup/Cukup Layak</b>	Sistem memerlukan perbaikan signifikan sebelum digunakan
<b>20%-39,99%</b>	<b>Tidak Layak</b>	Sistem tidak dapat diterima, perlu perancangan ulang
<b>0%-19,99%</b>	<b>Sangat Tidak Layak</b>	Sistem ditolak sepenuhnya

Berdasarkan Tabel 2.7, sistem dinyatakan diterima apabila persentase skor UAT secara keseluruhan berada pada kategori Baik/Layak atau di atasnya, yakni  $\geq 60\%$ .

## 2.10 Penelitian yang Berkaitan

Dalam penelitian ini, terdapat beberapa penelitian yang terkait sebagai referensi, di antaranya:

### 2.10.1 *Unlocking User Satisfaction: A DeLone & McLean IS Success Model Approach to IT Helpdesk Ticketing System Adoption*

Penelitian yang dilakukan oleh Niky Purnama Putra dan Astari Retnowardhani mengevaluasi faktor yang memengaruhi kepuasan pengguna pada aplikasi IT *Helpdesk Ticketing System* melalui model *DeLone & McLean Information System Success*. Data dikumpulkan melalui survei *online* kepada 231 responden dan dianalisis menggunakan PLS (*Structural Equation Model-Partial Least Square*) untuk menguji hubungan kualitas sistem, kualitas layanan, tingkat penggunaan, dan kepuasan pengguna. Hasilnya menunjukkan bahwa kualitas sistem, kualitas layanan, serta tingkat penggunaan berpengaruh signifikan terhadap kepuasan pengguna. Temuan ini memperkuat argumen bahwa perbaikan kinerja aplikasi dan mutu layanan dukungan (*support*) merupakan faktor kunci yang mendorong keberlanjutan penggunaan layanan digital berbasis *helpdesk* [5].

### 2.10.2 *Analyzing User Satisfaction with the Helpdesk System of PT. Matahari Department Store: An Evaluation Study*

Theresia Vania Davita Suyana dan Nilo Legowo menganalisis kepuasan pengguna sistem *helpdesk* di PT Matahari dan keterkaitannya dengan niat penggunaan berkelanjutan. Penelitian menggunakan kuesioner yang diisi oleh 280 responden dan dianalisis dengan SEM berbasis PLS, mencakup evaluasi validitas dan reliabilitas instrumen serta signifikansi hubungan antar variabel. Hasil analisis menegaskan bahwa kepuasan pengguna memiliki peran penting dalam mendorong niat penggunaan sistem secara berkelanjutan, sementara kualitas sistem dan persepsi manfaat/kemudahan menjadi faktor yang memperkuat kepuasan [6].

### **2.10.3 *Implementing online questionnaires and surveys by using mobile applications***

Penelitian ini dilakukan oleh Ramiz Salama, Huseyin Uzunboylu, dan Mohammad Abed El Muti yang menghasilkan sebuah *website* Q dan S untuk perusahaan dalam mengevaluasi karyawan dengan efisien dan cepat. Aplikasi dikembangkan menggunakan *NetBeans* IDE dengan teknologi seperti PHP dan WAMP Server. Sistem terintegrasi dengan *phpMyAdmin* untuk pengelolaan basis data. Pada penelitian ini juga membahas alat seperti *Google Forms*, *SurveyMonkey*, dan *Typeform*, dengan penekanan pada pendekatan *custom* dari platform Q dan S. Fitur utama pada aplikasi ini adalah membuat, mengelola, dan melihat survei dengan tanggal mulai/berakhir. Admin juga dapat menganalisis hasil survei secara *real-time* serta mengelola pengguna yang berpartisipasi dalam survei [8].

### **2.10.4 Penerapan *Waterfall* dalam Pengembangan Modul Survei Kepuasan Layanan pada Sistem Informasi Akademik Universitas Islam Negeri Antasari Banjarmasin**

Penelitian oleh Aulia Akbar Setyogomo merancang dan mengimplementasikan modul survei kepuasan layanan sebagai bagian dari sistem informasi akademik (modul tambahan pada sistem yang telah berjalan). Pengembangan dilakukan dengan model *Waterfall*. Pengujian sistem dilaksanakan secara berlapis, meliputi pengujian fungsional, pengujian *white-box* pada unit tertentu dengan teknik *basis path*, pengujian integrasi/validasi, uji kompatibilitas lintas peramban, serta uji *usability* menggunakan *System Usability Scale* (SUS). Hasil pengujian menunjukkan fungsi berjalan sesuai kebutuhan dan skor SUS=72, berada pada kategori baik/layak [4].

### **2.10.5 Sistem Informasi Survei Kepuasan Masyarakat dengan *Framework Laravel* pada Pelayanan Publik Pemerintah Daerah**

Penelitian yang dilakukan oleh Fadli Fadilillah, Faisal Amir, Riyanto, Rajimar Suhal Hasibuan, dan Syavira Indah Maryam mengembangkan sistem informasi survei kepuasan masyarakat berbasis web menggunakan *Laravel + MySQL* untuk menggantikan proses survei manual. Pengembangan dilakukan dengan model

*Waterfall* dan perancangan UML, kemudian sistem diuji menggunakan *blackbox testing* serta *User Acceptance Test (UAT)*. Hasil implementasi dilaporkan mempercepat pengolahan data dari 3–5 hari menjadi < 1 hari, meningkatkan akurasi data hingga 98%, dan menghasilkan kepuasan pengguna 85% pada UAT dengan 30 responden [3].

#### **2.10.6 *User Centered Method for Enhancing the Adoption of Software Systems in Palestine***

Rendahnya tingkat adopsi sistem perangkat lunak dinilai karena ketidaksesuaian metode pengembangan perangkat lunak yang digunakan dengan lingkungan kerja di Palestina. Yousef Awwad Daraghmi, Motaz Daadoo, dan Derar Eleyan mengusulkan metode pengembangan perangkat lunak baru yang berpusat pada pengguna (*user-centered*) yang mengintegrasikan *Participatory Design (PD)*, *Rapid Application Development (RAD)*, dan *Usability Engineering*. Metode ini diuji secara empiris dengan pengembangan sistem *E-Auction*, yang dirancang untuk proses lelang di pasar buah dan sayur di Palestina. Hasil evaluasi menunjukkan bahwa *E-Auction* dapat digunakan dan diterima dengan baik oleh pengguna. Penelitian ini menyimpulkan bahwa metode yang diusulkan efektif dalam meningkatkan adopsi sistem perangkat lunak di Palestina. Dengan mengintegrasikan PD, RAD, dan *Usability Engineering*, metode ini berhasil menghasilkan sistem yang sesuai dengan alur kerja dan lingkungan kerja pengguna akhir, sehingga meningkatkan kegunaan dan penerimaan sistem [11].

#### **2.10.7 *RAPD: Rapid and Participatory Application Development of Usable Systems During COVID19 Crisis***

Yousef-Awwad Daraghmi dan Eman-Yasser Daraghmi membahas tantangan pengembangan perangkat lunak pada masa Covid-19 yang perlu beralih ke sistem elektronik jarak jauh dan efisiensi, serta mudah dipahami dan memberikan kepuasan pengguna. Untuk menjawab tantangan tersebut, peneliti mengusulkan kerangka kerja *Rapid Application Participatory Development (RAPD)*. Selain itu, RAPD juga menambahkan pengujian kegunaan (*usability testing*) pada semua tahap desain untuk memvalidasi dan meningkatkan kegunaan sistem. Kerangka kerja

RAPD diuji secara empiris menggunakan metode Penelitian Aksi (*Action Research*) pada konteks kebutuhan mendesak akan sistem elektronik di Jericho *Central Vegetable Market*. Penelitian ini menyimpulkan bahwa RAPD merupakan kerangka kerja yang efektif untuk mengembangkan sistem yang dapat digunakan dan diterima selama krisis, seperti pandemi COVID-19. RAPD memungkinkan pengembangan yang cepat, partisipasi pengguna, dan pengujian kegunaan, yang semuanya berkontribusi pada keberhasilan penerapan sistem [12].

#### **2.10.8 *A digital Cohort Analysis of Consumers' Mobile Banking App Experience***

Penelitian yang dilakukan oleh Neeru Sharma membahas tentang perbedaan *user experience* aplikasi *mobile banking* antara *digital natives* (DN) dan *digital immigrants* (DI). DN adalah kelompok individu yang lahir setelah era digital, sedangkan DI ialah kelompok yang lahir sebelum era digital. Penelitian ini menggunakan metode kuantitatif dengan partisipan sebanyak 215 DN dan 203 DI di Australia yang dianalisis menggunakan teknik *partial least squares structural equation modeling* (PLS-SEM). Hasil penelitian menunjukkan bahwa terdapat perbedaan signifikan pada *user experience* aplikasi *mobile banking*. DN lebih mementingkan pengalaman pragmatis dan emosional, sedangkan DI lebih mementingkan kemudahan pengguna. Penelitian ini memberikan wawasan tentang pemahaman perbedaan pengalaman pengguna untuk meningkatkan kepuasan dan retensi pelanggan [7].

#### **2.10.9 *Risk-based Testing: Risk Analysis Fundamentals and Metrics for Software Testing Including a Financial Application Case Study***

Ståle Amland menerapkan *Risk-Based Testing* (RBT) pada proyek pengujian sistem perbankan. sebagai pendekatan untuk memfokuskan aktivitas pengujian pada fungsi/fitur yang memiliki risiko paling tinggi, terutama ketika waktu dan sumber daya pengujian terbatas. Penelitian ini menurunkan metrik risiko berupa *Risk Exposure* (RE) yang memadukan dua komponen utama, yaitu probabilitas terjadinya kesalahan pada suatu fungsi dan besarnya biaya/dampak apabila kesalahan tersebut terjadi pada lingkungan operasional. Penilaian risiko dilakukan terhadap fungsi/fitur yang disusun dalam *test item tree*; tim kemudian menetapkan

strategi risiko, indikator probabilitas dan biaya, serta menyusun daftar prioritas fungsi yang akan diuji. Setelah prioritas terbentuk, pengujian diposisikan sebagai upaya mitigasi risiko [38].

Berikut ini merupakan ringkasan dari penelitian yang berkaitan. Tabel 2.8 merupakan ringkasan dari penelitian terkait kepuasan pengguna.

Tabel 2.8 Penelitian Terkait Kepuasan Pengguna pada sistem *helpdesk*

Peneliti	Objek	Hasil	Keterkaitan/Gap
Niky Purnama Putra dan Astari Retnowardhani [5]	Evaluasi faktor yang memengaruhi kepuasan pengguna dan penggunaan berkelanjutan pada aplikasi IT <i>helpdesk ticketing</i> .	Kualitas sistem, kualitas layanan, dan tingkat penggunaan sistem berpengaruh signifikan terhadap kepuasan pengguna. Kepuasan pengguna memiliki peran penting dalam mendorong penggunaan berkelanjutan.	Studi berfokus pada evaluasi persepsi pengguna terhadap sistem yang sudah berjalan. Penelitian tersebut belum membahas perancangan/implementasi modul survei kepuasan yang terintegrasi dengan alur tiket ( <i>post-service feedback</i> ) maupun penggabungan data operasional-survei untuk analisis pengelompokan/periode.
Theresia Vania Davita Suyana dan Nilo Legowo [6]	Evaluasi kepuasan pengguna dan niat penggunaan berkelanjutan pada sistem <i>helpdesk</i> perusahaan.	<i>Perceived Usefulness</i> (PU) dan <i>System Quality</i> (SQ) berpengaruh signifikan pada kepuasan; <i>Perceived Ease of Use</i> (PEU) lebih lemah. <i>Continued Use Intention</i> (CUI) berkorelasi kuat dengan kepuasan.	Penelitian bersifat evaluatif (mengukur kepuasan pada sistem yang sudah ada) dalam konteks korporasi. Penelitian belum membahas desain sistem integrasi survei kepuasan sebagai bagian dari siklus penanganan layanan berbasis tiket, maupun penyajian analisis pengelompokan/periode untuk <i>monitoring</i> .

Tabel 2.9 merupakan ringkasan penelitian terkait dengan survei *online*/digital.

Tabel 2.9 Penelitian Terkait Survei *Online*/Digital

Peneliti	Objek	Hasil	Keterkaitan/Gap
Ramiz Salama, Huseyin Uzunboylu, dan Mohammad Abed El Muti [8]	Implementasi kuesioner dan survei <i>online</i> berbasis <i>website</i> .	Situs web kuesioner dan survei <i>online</i> yang dapat digunakan perusahaan untuk mengevaluasi kinerja, melakukan pemungutan suara, atau mengumpulkan umpan balik dari karyawan.	Studi berfokus pada platform survei generik; tidak memosisikan survei sebagai bagian dari evaluasi layanan. Penelitian tersebut tidak membahas analisis pengelompokan/periode ( <i>cohort</i> ) berbasis waktu/retensi atau penggabungan data operasional survei. Pengujian perangkat lunak tidak dibahas secara rinci.
Aulia Akbar Setyogomo [4]	Pengembangan modul survei kepuasan layanan sebagai modul tambahan pada Sistem Informasi Akademik (Siakad).	Modul survei kepuasan layanan sebagai bagian dari Sistem Informasi Akademik (SIA/Siakad) yang terintegrasi pada sistem yang sudah berjalan.	Integrasi modul dilakukan pada konteks sistem akademik. Pelaporan hasil survei belum dikembangkan menjadi analisis pengelompokan/periode ( <i>cohort</i> ) untuk <i>monitoring</i> partisipasi/ <i>usage</i> dari waktu ke waktu. Pendekatan pengembangan <i>Waterfall</i> berbeda dengan pendekatan <i>iteratif</i> cepat (RAPD) yang digunakan pada penelitian ini.
Fadli Fadilillah, Faisal Amir, Riyanto, Rajimar Suhail Hasibuan, dan Syavira Indah Maryam [3]	Pengembangan sistem survei kepuasan masyarakat berbasis web untuk layanan publik.	Sistem informasi survei kepuasan masyarakat berbasis web ( <i>Laravel</i> ) yang menyediakan kuesioner digital, pengolahan/rekap otomatis, serta <i>dashboard</i> pelaporan.	Sistem diposisikan sebagai platform survei dan pelaporan umum; keterkaitan survei dengan <i>workflow</i> layanan tidak menjadi fokus. Analisis yang disajikan berorientasi rekap/ <i>dashboard</i> ; belum diarahkan pada <i>monitoring</i> berbasis pengelompokan ( <i>cohort</i> ). Penelitian tidak membahas strategi prioritas pengujian saat banyak fungsi dengan sumber daya terbatas.

Tabel 2.10 berikut merupakan ringkasan dari penelitian yang berkaitan dengan penerapan metode *Rapid and Participatory Application Development* (RAPD).

Tabel 2.10 Penelitian Terkait RAPD

Peneliti	Objek	Hasil	Keterkaitan/Gap
Yousef Awwad Daraghmi, Motaz Daadoo, dan Derar Eleyan [11]	Metode <i>user-centered</i> untuk meningkatkan adopsi sistem melalui integrasi RAD dan <i>participatory/usability engineering</i> .	Metode pengembangan berpusat pengguna yang mengintegrasikan RAD dan <i>participatory design</i> , serta contoh evaluasi <i>usability</i> pada studi kasus aplikasi.	Studi ini memberikan prinsip pengembangan berpusat pengguna ( <i>user-centered</i> ) serta contoh evaluasi <i>usability</i> pada studi kasus aplikasi yang berbeda. Pada penelitian ini, prinsip keterlibatan pengguna diadaptasi melalui iterasi prototipe/validasi kebutuhan dan pengujian penerimaan (UAT) pada aplikasi <i>helpdesk</i> survei.
Yousef-Awwad Daraghmi dan Eman-Yasser Daraghmi [12]	Kerangka pengembangan cepat dan partisipatif untuk menghasilkan sistem yang <i>usable</i> dalam kondisi kebutuhan mendesak.	Kerangka RAPD yang menggabungkan RAD dan <i>participatory design</i> , disertai penempatan aktivitas <i>prototyping</i> dan <i>usability</i> testing pada beberapa tahap pengembangan untuk memastikan sistem <i>usable</i> .	Kerangka RAPD bersifat umum dan dapat diterapkan pada berbagai domain. Pada penelitian ini, RAPD dioperasionalkan menjadi tahapan kerja yang sesuai konteks pengembangan aplikasi <i>helpdesk</i> terintegrasi survei di UPA TIK. Dengan demikian, kontribusi rujukan ini pada penelitian adalah sebagai landasan metodologis, sementara perincian artefak dan indikator evaluasi ditetapkan berdasarkan kebutuhan sistem yang dibangun.

Tabel 2.11 berikut ini merupakan ringkasan dari penelitian yang berkaitan dengan *cohort analysis*.

Tabel 2.11 Penelitian Terkait *Cohort Analysis*

Peneliti	Objek	Hasil	Keterkaitan/Gap
Neeru Sharma [7]	Analisis perbedaan pengalaman, kepuasan, dan niat penggunaan berkelanjutan antara <i>cohort</i> pengguna ( <i>digital natives vs digital immigrants</i> ).	Model analisis dan temuan perbandingan antar kelompok <i>cohort</i> (berdasarkan karakteristik pengguna) terkait pengalaman aplikasi, kepuasan, dan niat penggunaan berkelanjutan.	Definisi <i>cohort</i> pada studi tersebut berbasis karakteristik pengguna ( <i>natives vs immigrants</i> ), bukan <i>cohort</i> berbasis periode waktu untuk <i>monitoring</i> retensi/partisipasi. Sedangkan penelitian ini menggunakan <i>cohort</i> sebagai fitur analisis untuk pelaporan <i>monitoring</i> ( <i>usage/retensi</i> sederhana) tanpa interpretasi mendalam.

Tabel 2.12 berikut ini merupakan ringkasan dari penelitian yang berkaitan dengan *Risk-based Testing*.

Tabel 2.12 Penelitian Terkait Risk-based Testing

Peneliti	Objek	Hasil	Keterkaitan/Gap
Ståle Amland [38]	Penerapan RBT pada proyek pengujian perangkat lunak (studi kasus sistem perbankan) dengan fokus pada pemetaan risiko per fungsi/fitur.	Menghasilkan metrik dan prosedur operasional untuk prioritas pengujian berbasis risiko, yaitu daftar fungsi/fitur prioritas yang menjadi fokus pengujian ketika sumber daya terbatas.	Menjadi dasar akademik untuk menjelaskan pengujian yang terarah dan pembatasan cakupan pengujian pada fungsi tertentu. Gap utama pada penelitian ini adalah konteks domain yang berbeda (perbankan dan <i>helpdesk</i> ), sehingga indikator probabilitas/dampak perlu disesuaikan dengan karakteristik modul/fungsi pada sistem yang dibangun.

Penelitian terdahulu mengenai sistem *helpdesk* umumnya berfokus pada kepuasan pengguna dan faktor-faktor yang memengaruhinya, seperti kualitas sistem, kemudahan penggunaan, dan manfaat yang dirasakan. Temuan tersebut menunjukkan bahwa keberhasilan *helpdesk* tidak hanya ditentukan oleh fitur, tetapi juga kualitas implementasi dan pengalaman pengguna. Sementara itu, penelitian mengenai survei/kuesioner *online* memperlihatkan bahwa digitalisasi survei mampu meningkatkan efisiensi pengumpulan serta pengolahan data. Namun, sebagian penelitian masih memosisikan survei sebagai sistem yang berdiri sendiri, sehingga keterkaitan survei kepuasan dengan alur layanan operasional seperti *helpdesk* belum menjadi fokus utama.

Dari sisi metodologi, pendekatan *user-centered* dan kerangka RAPD menegaskan pentingnya iterasi cepat, keterlibatan pengguna, serta validasi penerimaan untuk menghasilkan sistem yang sesuai kebutuhan. Literatur mengenai *risk-based testing* juga menyediakan dasar untuk memprioritaskan pengujian pada area berisiko tinggi ketika sumber daya terbatas. Sementara itu, konsep *cohort analysis* turut menguatkan gagasan bahwa pengelompokan pengguna dapat dimanfaatkan untuk melihat pola keterlibatan/aktivitas berdasarkan kelompok tertentu.

Dengan demikian, celah yang teridentifikasi adalah belum banyak penelitian yang mengintegrasikan *helpdesk* berbasis tiket, survei kepuasan, dan fitur analisis berbasis pengelompokan dalam satu kesatuan sistem, serta mendokumentasikan pengujian yang dilakukan secara terarah. Penelitian ini mengisi celah tersebut melalui perancangan dan implementasi aplikasi terintegrasi untuk UPA TIK Universitas Lampung yang menyediakan survei kepuasan dan fitur analisis pengelompokan berdasarkan periode tertentu. Sesuai batasan penelitian, hasil analisis tersebut tidak diarahkan pada interpretasi mendalam, prediksi, maupun evaluasi retensi secara mendalam. Pengujian sistem dilakukan melalui pengujian unit, pengujian fungsional, dan penerimaan pengguna, dengan cakupan yang diprioritaskan menggunakan prinsip *risk-based testing*, sehingga keterbatasan waktu dan sumber daya tetap dapat menghasilkan pengujian yang terukur.



## 3.2 Alat dan Bahan

### 3.2.1 Alat Penelitian

Pada penelitian ini menggunakan beberapa alat untuk mendukung dan menunjang setiap tahapan penelitian. Adapun alat yang digunakan berupa perangkat lunak dan perangkat keras. Tabel 3.2 merupakan daftar perangkat lunak yang digunakan pada penelitian ini.

Tabel 3.2 Perangkat Lunak

No.	Perangkat	Spesifikasi	Kegunaan
1.	<i>Visual Studio Code</i>	V 1.109.4	IDE/editor untuk pengembangan aplikasi
2.	<i>PostgreSQL</i>	V 18.1-2	DBMS untuk penyimpanan data aplikasi
3.	<i>Flutter</i>	V 3.38.9	<i>Framework</i> pengembangan aplikasi <i>client</i>
4.	<i>Dart</i>	V 3.10.8	Bahasa pemrograman untuk pengembangan <i>frontend</i>
5.	<i>Go</i>	V 1.25.1	Bahasa pemrograman untuk pengembangan <i>backend</i>
6.	<i>Microsoft Visio</i>	V 2411	Perangkat lunak pembuatan Diagram
7.	<i>Firebase CLI</i>	V15.6.0	<i>Hosting</i> untuk <i>deployment</i> web admin sebagai lingkungan uji ( <i>staging/testing</i> )
8.	<i>Firebase Cloud Messaging (FCM)</i>	-	Layanan pengiriman notifikasi <i>push</i> ke perangkat ( <i>Android</i> )
9.	<i>Heroku</i>	-	<i>Hosting</i> untuk <i>deployment backend/API</i> sebagai lingkungan uji ( <i>staging/testing</i> )
10.	<i>Google Chrome</i>	V 145.0.7632.109	Perangkat lunak pengujian <i>website</i> admin
11.	<i>Postman</i>	V 12.1.1	Perangkat lunak pengujian <i>request API Endpoint</i>
12.	<i>Github</i>	-	<i>Version control system</i> dan sumber <i>automatic deploy backend</i>
13.	<i>Cloudflare</i>	-	Konfigurasi <i>custom domain</i>

Tabel 3.3 merupakan daftar perangkat keras yang digunakan pada penelitian ini.

Tabel 3.3 Perangkat Keras

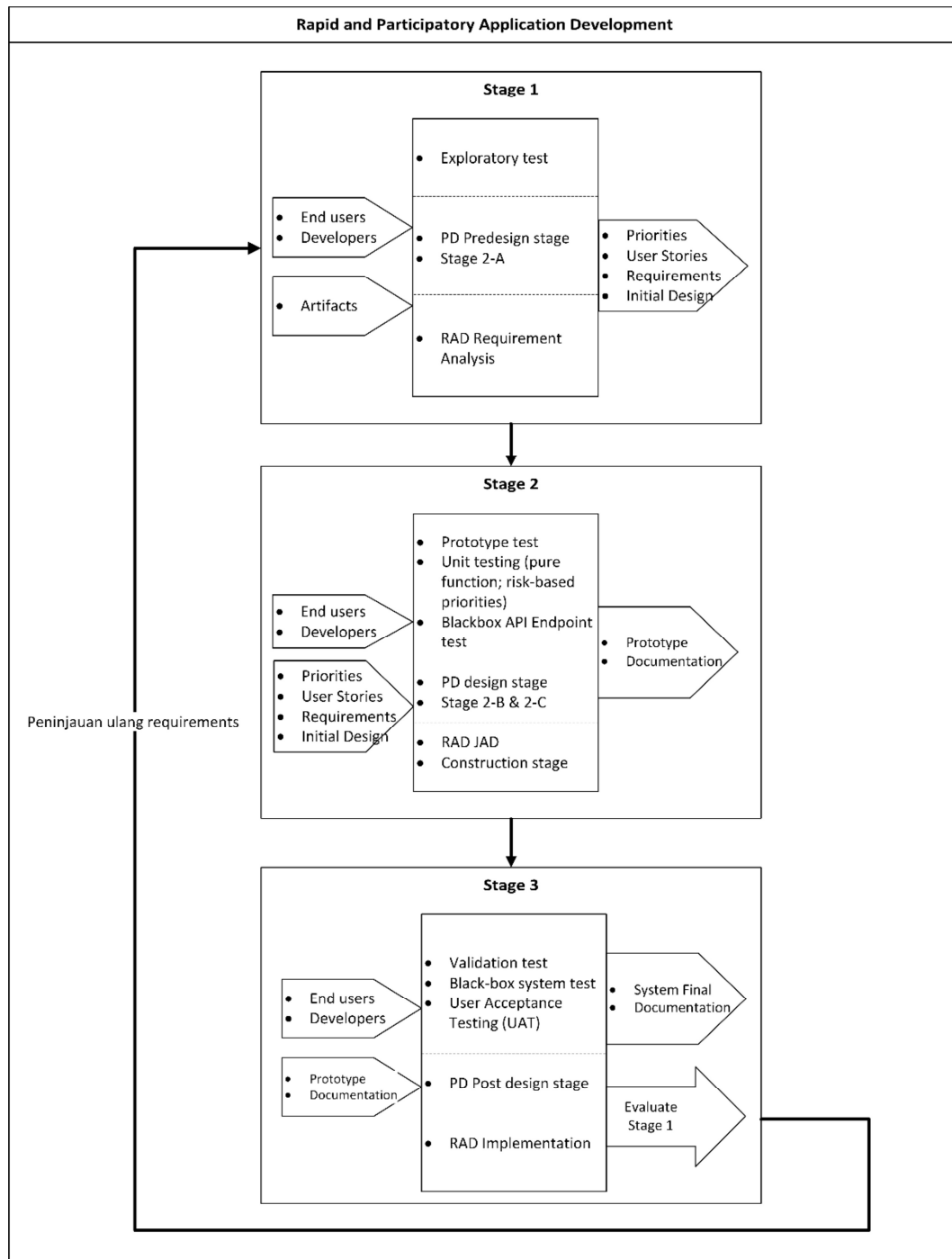
No.	Perangkat	Spesifikasi	Kegunaan
1.	Laptop	AMD <i>Ryzen</i> 5-3500U, <i>Radeon Vega</i> 8, RAM 8 GB, Sistem Operasi Windows 11	Perangkat pembuatan dan pengujian aplikasi
2.	<i>Smartphone Android</i>	<i>Android Version</i> 12, <i>Exynos</i> 9611 <i>processor</i> , RAM 4 GB	Perangkat pengujian Aplikasi

### 3.2.2 Bahan Penelitian

Pada penelitian ini menggunakan beberapa dokumen pendukung sebagai informasi yang dapat menunjang penelitian, di antaranya:

1. Beberapa penelitian terdahulu yang relevan.
2. Berbagai buku dan modul bacaan sebagai referensi utama.
3. Jurnal nasional dan internasional sebagai acuan penelitian.

### 3.3 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian [12].

Gambar 3.1 menunjukkan metodologi pengembangan sistem yang digunakan dalam penelitian ini, yaitu *Rapid and Participatory Application Development* (RAPD). Metode ini terdiri atas tiga *stage* utama yang saling berkesinambungan,

yaitu *Stage 1*, *Stage 2*, dan *Stage 3*. Pada penelitian ini, RAPD digunakan sebagai kerangka pengembangan untuk menghasilkan sistem secara cepat dengan tetap melibatkan pengguna dalam proses perancangan dan evaluasi sistem.

*Stage 1* berfokus pada perumusan kebutuhan dan rancangan awal sistem. Pada tahap ini, pengembang dan pengguna membangun kesepahaman mengenai tujuan, ruang lingkup, prioritas pengembangan, kebutuhan sistem, serta rancangan awal yang akan menjadi dasar implementasi. Kegiatan pada tahap ini didukung oleh observasi, diskusi dengan pemangku kepentingan, serta penelaahan artefak yang relevan untuk memperoleh kebutuhan awal sistem. Selain itu, *exploratory test* dilakukan bersama pengguna untuk memvalidasi kesesuaian kebutuhan, alur proses, dan rancangan awal sebelum pengembangan dilanjutkan ke tahap berikutnya.

*Stage 2* berfokus pada pengembangan prototipe fungsional secara cepat dan *iteratif*. Pada tahap ini dilakukan penyempurnaan kebutuhan dan desain, implementasi fitur, serta dokumentasi prototipe yang telah dikembangkan. *Prototype test* dilakukan secara bertahap bersama perwakilan pengguna untuk memperoleh umpan balik terhadap fungsi dan alur sistem, sehingga perbaikan dapat dilakukan pada iterasi berikutnya. Selain itu, pengujian teknis juga dilakukan oleh pengembang melalui *blackbox API endpoint test* untuk memverifikasi kesesuaian respons API terhadap spesifikasi yang telah ditentukan. Prioritas pengujian ditetapkan menggunakan pendekatan *Risk-Based Testing* (RBT), sedangkan unit testing dilakukan secara terbatas pada fungsi murni (*pure function*) untuk memverifikasi logika kritis pada komponen inti sistem.

*Stage 3* berfokus pada integrasi akhir sistem, *deployment* pada lingkungan *staging*, serta validasi sistem. Pada tahap ini, sistem yang telah dikembangkan dilakukan *Validation test* bersama perwakilan pengguna untuk memastikan semua kebutuhan telah terpenuhi. Fungsional sistem diuji secara menyeluruh melalui *blackbox system test* untuk memastikan integrasi antara *frontend* dan *backend* berjalan dengan baik secara *end-to-end*. Selain itu, menggunakan *User Acceptance Testing* (UAT)

dilakukan dengan melibatkan pengguna akhir secara langsung untuk menilai tingkat penerimaan sistem dan memastikan bahwa sistem yang dikembangkan telah sesuai dengan kebutuhan pengguna. Melalui tiga *stage* tersebut, RAPD pada penelitian ini tidak hanya mendukung pengembangan sistem secara cepat, tetapi juga memastikan bahwa sistem yang dihasilkan telah melalui proses evaluasi dan validasi pada setiap tahap pengembangannya.

Penerapan RAPD pada penelitian ini disesuaikan dengan konteks pengembangan sistem serta artefak yang tersedia. Hal ini dibenarkan oleh [13] yang menunjukkan bahwa metode RAPD dapat dimodifikasi berdasarkan konteks penelitian dan ketersediaan artefak, sebagaimana yang dilakukan pada penelitian tersebut dengan menghapus *comparison test* karena sistem desktop yang sudah ada dapat langsung dijadikan artefak elektronik. Pada penelitian ini, penyesuaian dilakukan dengan menggantikan *prototype test* berbasis rancangan antarmuka statis menjadi pengujian menggunakan *prototype fungsional*, serta menambahkan pengujian bertingkat, yaitu *blackbox API endpoint test* dan *unit testing* sebagai bentuk verifikasi teknis. Konsep *prototype* yang digunakan dalam penelitian ini mengacu pada pendekatan *working prototype* atau *evolutionary prototype* sebagaimana yang digunakan oleh [11] yang menggunakan *prototype* sebagai sistem yang diimplementasikan sebagian dan terus disempurnakan melalui umpan balik pengguna pada setiap iterasi, bukan sekadar representasi visual statis. Selain itu, pada *stage 3* tidak hanya dilakukan *validation test*. Pada penelitian ini juga menambahkan pengujian *blackbox system test end-to end* sebelum dilakukan penilaian dan penerimaan sistem oleh pengguna akhir menggunakan UAT.

### 3.3.1 *Stage 1*

#### **Partisipan:**

Perwakilan pengguna dan pengembang.

#### **Prasyarat:**

Tujuan penelitian dan ruang lingkup awal, serta artefak pendukung berupa kondisi layanan *helpdesk* yang sedang berjalan dan dokumen referensi yang relevan.

**Aktivitas:**

## 1. Diskusi Terbuka dengan Pemangku Kepentingan

Observasi dan diskusi dilaksanakan pada 18 September 2025 bertempat di UPA TIK Universitas Lampung. Diskusi dilakukan dalam bentuk tanya jawab tanpa daftar pertanyaan baku, namun diarahkan pada beberapa topik utama, di antaranya: (a) mekanisme pengumpulan data kepuasan layanan yang saat ini berjalan; (b) hambatan dan kendala yang dihadapi dalam proses survei kepuasan yang ada; (c) fitur dan kapabilitas sistem *helpdesk* yang telah berjalan; (d) kebutuhan integrasi antara sistem *helpdesk* dengan mekanisme survei kepuasan; serta (e) ekspektasi terhadap sistem yang akan dikembangkan. Diskusi ini dilakukan dengan Ibu Dr. Eng. Ir. Mardiana, S.T., M.T., I.P.M. selaku Kepala UPA TIK Universitas Lampung, serta Supervisor *Helpdesk* UPA TIK, yaitu Bapak M. Iqbal Parabi, S.Si., M.T.

Dari observasi dan diskusi tersebut diperoleh temuan bahwa: (a) pengumpulan data kepuasan saat ini masih dilakukan secara manual melalui penyebaran kuesioner fisik dengan rekapitulasi yang juga dilakukan secara manual, sehingga menimbulkan kendala efisiensi, jangkauan responden, dan validitas data; (b) sistem *helpdesk* UPA TIK sudah berbasis *website* dengan fitur *ticketing*, tetapi belum terintegrasi dengan mekanisme survei kepuasan, sehingga data operasional layanan dan data persepsi pengguna belum dapat dikaitkan secara langsung; (c) integrasi antara *helpdesk* dan survei kepuasan akan memungkinkan umpan balik dikaitkan langsung dengan layanan yang benar-benar diterima pengguna, sehingga hasil survei lebih relevan dan dapat dipertanggungjawabkan sebagai bahan evaluasi instansi.

## 2. Perumusan kebutuhan awal

Kebutuhan awal dituliskan dalam bentuk *user stories*, *use case* utama, dan prioritas pengembangan berdasarkan dependensi antar fitur. *User stories* digunakan untuk menggambarkan fungsi yang bernilai bagi pengguna dari sudut pandang peran (*role*). *User story* tidak hanya sebagai kalimat kebutuhan, tetapi juga sebagai pemicu diskusi (*conversation*) dan dasar verifikasi melalui *acceptance tests* yang menjelaskan kapan suatu kebutuhan dinyatakan terpenuhi. Dengan demikian,

perumusan *user stories* pada penelitian ini digunakan sebagai cara untuk menyepakati kebutuhan bersama *stakeholder*; mengelompokkan kebutuhan berdasarkan prioritas, serta mengarahkan pengembangan fitur secara bertahap sesuai ruang lingkup penelitian [41]. Untuk memperjelas batas pemenuhan kebutuhan, setiap *user story* dilengkapi *acceptance criteria* yang dituliskan dalam format **Given–When–Then** untuk menyatakan kondisi awal (**Given**), aksi (**When**), dan hasil yang diharapkan (**Then**). Selain membantu mengurangi ambiguitas, penulisan kriteria dalam bentuk skenario juga memudahkan pelacakan kebutuhan ke aktivitas pengujian, karena setiap skenario dapat diturunkan menjadi kasus uji fungsional yang memverifikasi perilaku sistem sesuai kebutuhan [42]. Sementara itu, penentuan prioritas pada setiap *stage* dilakukan berdasarkan dependensi antar aktivitas dalam proses pengembangan sistem. Aktivitas dengan prioritas 1 merupakan aktivitas yang harus dilakukan lebih dahulu karena menjadi dasar bagi aktivitas berikutnya, sedangkan aktivitas dengan prioritas 2, 3, 4 dan 5 dilakukan setelah komponen utama pada *stage* tersebut tersedia atau telah tervalidasi.

### 3. Perancangan Awal Sistem

Perancangan awal dilakukan sebagai acuan *prototyping*, meliputi arsitektur sistem, alur proses utama, rancangan data, *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *Entity Relationship Diagram* (ERD).

### 4. *Exploratory Test*

*Exploratory test* dilakukan untuk memvalidasi rancangan awal dan kebutuhan yang telah diidentifikasi bersama pengguna, memastikan kesesuaian arah pengembangan sebelum masuk ke *Stage 2*.

#### **Hasil yang Diharapkan:**

1. Kesepakatan ruang lingkup dan rencana kegiatan/iterasi (prioritas).
2. Ringkasan kebutuhan pengguna, termasuk spesifikasi kebutuhan fungsional dan non-fungsional, serta *user stories*.

3. Desain awal sistem (*initial design*) yang mencakup rancangan arsitektur sistem, *use case diagram*, *activity diagram*, *sequence diagram*, dan *entity relational diagram* (ERD).

### 3.3.2 Stage 2

#### **Partisipan:**

Perwakilan pengguna akhir dan pengembang.

#### **Prasyarat:**

1. Ringkasan kebutuhan pengguna, termasuk spesifikasi kebutuhan fungsional dan non-fungsional, *user stories*, serta prioritas dari *stage* 1.
2. Desain awal sistem, termasuk rancangan arsitektur sistem, *use case diagram*, *activity diagram*, *sequence diagram*, dan *entity relational diagram* (ERD).

#### **Aktivitas:**

1. Konfirmasi kebutuhan

Konfirmasi kebutuhan kepada pengguna dilakukan untuk memastikan kesesuaian skenario utama.

2. Pengembangan *Prototype* Fungsional

Pengembangan *Prototype* Fungsional sebagai sistem yang langsung dapat dijalankan secara cepat dan *iteratif*, yang mencakup penyempurnaan desain, implementasi fitur, serta penyusunan kasus uji awal.

3. *Prototype Test*

Pengujian *prototype* fungsional bersama perwakilan pengguna secara informal menggunakan prototipe fungsional sistem, yaitu sistem yang telah diimplementasikan sebagian dalam bentuk kode yang dapat dijalankan, bukan sekadar rancangan antarmuka statis. Umpan balik yang diperoleh digunakan sebagai dasar perbaikan pada iterasi berikutnya.

#### 4. Unit testing

Pengujian unit dilakukan pada *pure function* menggunakan pendekatan *Risk-Based Testing* (RBT) yang memprioritaskan pengujian berdasarkan tingkat risiko agar sumber daya pengujian dapat difokuskan pada komponen yang paling kritis terhadap keberhasilan sistem. Penilaian risiko mengacu pada parameter yang dijelaskan oleh ISTQB, yaitu *Impact* (I) dan *Likelihood* (L) yang masing-masing dinilai pada skala 1–5. Skor risiko dihitung sebagai hasil perkalian kedua parameter tersebut menggunakan perhitungan pada persamaan (2.1). Klasifikasi tingkat risiko menggunakan *L-Type Risk Matrix* 5×5 yang didefinisikan pada Tabel 3.4 [43].

Tabel 3.4 Klasifikasi Risiko

Klasifikasi	Skor Risiko
<i>High</i>	15, 16, 20, 25
<i>Medium</i>	8, 9, 10, 12
<i>Low</i>	1, 2, 3, 4, 5, 6

Berdasarkan klasifikasi tersebut, *Pure function* dengan level *High* dan *Medium* diuji melalui *Whitebox Unit Test*, sedangkan *pure function* dengan level *Low* serta seluruh fungsi non-pure cukup diuji melalui *Blackbox Testing* dan UAT. *Pure function* dalam penelitian ini didefinisikan sebagai fungsi yang deterministik, tidak mengakses basis data, jaringan, *file*, maupun *context* HTTP, tidak menghasilkan *side effect*, dan dapat diuji tanpa *mock* eksternal.

#### 5. Blackbox API Endpoint Test

*Prototype test* dilakukan melalui pengujian *blackbox API endpoint* menggunakan *Postman Collection Runner* terhadap seluruh *endpoint backend* sistem. *Decision Table Testing* dipilih untuk menurunkan kasus uji karena mampu menangkap kombinasi kondisi masukan (*input*) dan perilaku sistem secara sistematis, sehingga cocok digunakan untuk menguji fungsi yang melibatkan hubungan logis antara dua kondisi atau lebih. Setiap kasus uji diturunkan dari *acceptance criteria* pada masing-masing *user story*, sehingga setiap *test case* memiliki keterlacakan (*traceability*) yang jelas terhadap skenario yang diuji. Pengujian dilaksanakan secara *iteratif*, setiap kegagalan akan dianalisis akar masalahnya, diperbaiki, dan dieksekusi kembali hingga seluruh *assertion* lulus. Proses penurunan kasus uji

diawali dengan menyusun *decision table* seperti pada Tabel 2.5 yang mendefinisikan kombinasi kondisi (*conditions*) dan aksi yang dihasilkan (*actions*) untuk setiap *user story*.

#### 6. Persiapan lingkungan *deployment*

Persiapan lingkungan *deployment* dilakukan untuk memastikan *prototype* fungsional dapat dijalankan pada lingkungan pengujian yang mendekati produksi. Tabel 3.5 merupakan spesifikasi kebutuhan server dan layanan *deployment* yang akan digunakan pada lingkungan *staging*.

Tabel 3.5 Spesifikasi Kebutuhan Server dan Layanan Deployment

Komponen	Layanan / Paket	Keterangan
<b>Backend API</b>	Heroku Basic Dynos (region US)	<i>Container</i> Linux tanpa <i>sleep</i> otomatis. <i>Deployment</i> otomatis via <i>push</i> ke <i>branch</i> utama <i>GitHub</i> .
<b>Basis Data</b>	Heroku <i>Postgres Essential-0</i>	<i>PostgreSQL</i> terkelola. <i>Connection string</i> tersedia otomatis via <i>DATABASE_URL</i> saat <i>add-on</i> dipasang.
<b>Antarmuka Web Admin</b>	<i>Firebase Hosting (helpdesk-unila.web.app)</i>	<i>Hosting</i> statis <i>Flutter</i> web dengan <i>CDN</i> global Google. <i>Deploy</i> manual via <i>Firebase CLI</i> .
<b>Notifikasi Push</b>	<i>Firebase Cloud Messaging (FCM) - project helpdesk-unila</i>	<i>Backend</i> menggunakan <i>Service Account JSON Firebase Admin SDK</i> . Aplikasi <i>Flutter Android</i> menerima notifikasi <i>push</i> melalui <i>Firebase Cloud Messaging (FCM)</i> .

#### Hasil yang Diharapkan:

1. *Prototype* fungsional yang telah menjalankan alur utama sistem secara *end-to-end*.
2. Dokumentasi yang diperbarui, mencakup kebutuhan, rancangan, perubahan hasil iterasi, kasus uji awal, dan hasil uji *prototype* informal yang telah diperbaiki.
3. Hasil *blackbox API endpoint test* dan *unit test* pada fungsi murni sebagai bukti verifikasi *critical logic*.

### 3.3.3 Stage 3

#### **Partisipan:**

Tim pengembang, perwakilan pengguna, meliputi pengguna langsung (*end user*) dan pihak terkait.

#### **Prasyarat:**

1. Prototipe fungsional dari *Stage 2*.
2. Dokumentasi rancangan serta catatan perubahan.

#### **Kegiatan:**

1. Integrasi Sistem dan Deployment

Integrasi akhir dan penyempurnaan sistem menjadi sistem utuh, serta implementasi pada lingkungan server (*deployment*), untuk kebutuhan pengujian, meliputi integrasi *frontend (Flutter/Dart)*, *backend (Golang)*, dan basis data (*PostgreSQL*).

2. *Validation Test*

Peninjauan akhir kebutuhan oleh pengguna dan pengembang untuk memastikan seluruh persyaratan prioritas telah terakomodasi. Apabila ditemukan kebutuhan baru yang signifikan, proses dapat kembali ke *Stage 1* untuk evaluasi dan perencanaan ulang.

3. *Blackbox System Test*

*Validation test* melalui pengujian *blackbox system test* untuk menguji integrasi antara *frontend* dan *backend* secara *end-to-end*. Penurunan kasus uji pada tahap pengujian ini adalah menggunakan pendekatan *scenario-based testing*, yaitu menurunkan skenario uji berdasarkan *acceptance criteria* dari masing-masing *user story*.

4. *User Acceptance Testing (UAT)*

UAT dilaksanakan bersama perwakilan pengguna akhir untuk menilai tingkat penerimaan sistem dan memastikan bahwa sistem yang dikembangkan telah sesuai dengan kebutuhan pengguna. UAT dibagi menjadi dua kelompok, yaitu UAT

pengguna umum (mahasiswa, dosen, dan tenaga kependidikan) yang berinteraksi dengan aplikasi Android menggunakan aplikasi *android* dan UAT admin dari UPA TIK yang berinteraksi dengan antarmuka web admin melalui perangkat desktop/laptop. Sesi pengujian dilaksanakan secara tatap muka maupun melalui daring dengan didampingi pengembang pada setiap partisipasi pengujian, sehingga kendala yang muncul dapat segera ditangani. UAT dilakukan menggunakan Instrumen kuesioner yang disusun berdasarkan aspek fungsional sistem yang dikembangkan, mencakup aspek autentikasi, manajemen tiket, notifikasi, survei kepuasan, dan penilaian umum sistem. Perhitungan skor UAT menggunakan rumus yang didefinisikan pada Tabel 2.6 dengan kriteria penerimaan pada Tabel 2.7.

**Hasil yang Diharapkan:**

1. Sistem yang berjalan pada server atau lingkungan uji (*staging/hosting*) dan dapat diakses sesuai kebutuhan pengujian.
2. Hasil pengujian sistem (*blackbox system test*), hasil UAT, serta dokumentasi final yang mencakup kebutuhan, rancangan, pengujian, dan hasil validasi.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan yang menjawab rumusan masalah penelitian, maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi survei kepuasan pengguna berbasis *mobile* berhasil dirancang dan dikembangkan secara terintegrasi dengan sistem *helpdesk* UPA TIK Universitas Lampung. Sistem memungkinkan pengguna terdaftar mengajukan tiket layanan dan mengisi survei kepuasan yang terkait langsung dengan layanan yang telah diterima, sehingga umpan balik yang terkumpul lebih terstruktur dan relevan sebagai bahan evaluasi bagi instansi. Berdasarkan hasil *black box system test* terhadap 38 skenario yang seluruhnya lolos, serta *User Acceptance Testing* terhadap 27 responden dengan persentase penerimaan 92,13% ber kriteria Sangat Baik, sistem dinyatakan memenuhi kebutuhan fungsional dan layak digunakan.
2. Fitur kustomisasi kuesioner berhasil dirancang dan diimplementasikan dengan mekanisme yang memungkinkan pengguna mengisi survei hanya untuk layanan yang telah selesai ditangani. Selain itu, admin dapat membuat *template* survei dengan berbagai tipe pertanyaan, meliputi pertanyaan dengan jawaban ya/tidak, skala *Likert* 3 titik, 4 titik, 5 titik, pilihan ganda, dan jawaban bebas, serta mengaitkannya dengan kategori layanan tertentu.
3. Fitur rekapitulasi hasil survei dan analisis *cohort* berhasil dirancang dan diimplementasikan. Sistem menyajikan ringkasan hasil survei secara menyeluruh berdasarkan filter periode dan kategori layanan, serta menampilkan visualisasi *cohort* dalam bentuk matriks kelompok–periode untuk memantau

dinamika kepuasan pengguna dari waktu ke waktu sebagai bahan evaluasi layanan.

## 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan serta masukan yang diperoleh selama proses pengujian dan validasi, terdapat beberapa saran untuk pengembangan sistem lebih lanjut:

1. Penelitian selanjutnya dapat mengkaji strategi integrasi sistem survei kepuasan dengan sistem *helpdesk* yang sudah beroperasi melalui mekanisme API atau sinkronisasi data. Penelitian ini membangun replikasi manajemen tiket secara terpisah, sehingga studi tentang pendekatan integrasi langsung akan menghasilkan temuan yang lebih aplikatif dan dapat diadopsi tanpa menggantikan alur kerja yang sudah berjalan.
2. Penelitian selanjutnya dapat menyelidiki pengaruh mekanisme kewajiban pengisian survei terhadap tingkat respons dan akurasi data kepuasan. Perbandingan antara skema pengisian sukarela dan skema yang didasarkan pada insentif atau pembatasan akses dapat menjadi fokus penelitian lanjutan.
3. Penelitian selanjutnya dapat mengeksplorasi pengaruh kualitas antarmuka pengguna terhadap tingkat adopsi sistem layanan berbasis mobile di institusi pendidikan tinggi. Variabel independen yang dapat dipertimbangkan meliputi konsistensi visual, kemudahan navigasi, dan daya tarik estetika antarmuka, dengan pengaruhnya diukur melalui skor UAT dan tingkat kepuasan pengguna secara umum.
4. Penelitian selanjutnya perlu membahas mekanisme pencegahan spam pada API, seperti penerapan *rate limiting* guna membatasi jumlah *request* yang dapat dilakukan dalam rentang waktu yang sama atau berdekatan, sehingga dapat meningkatkan keamanan dan stabilitas sistem secara keseluruhan.

## **DAFTAR PUSTAKA**

## DAFTAR PUSTAKA

- [1] S. M. Dam and T. C. Dam, “Relationships between Service Quality, Brand Image, Customer Satisfaction, and Customer Loyalty,” *Journal of Asian Finance, Economics and Business*, vol. 8, no. 3, pp. 585–593, Mar. 2021, doi: 10.13106/jafeb.2021.vol8.no3.0585.
- [2] *Undang-Undang Republik Indonesia Nomor 25 Tahun 2009 tentang Pelayanan Publik*. Indonesia: LN. 2009/ No.112 , TLN NO. 5038, LL SETNEG : 45 HLM, 2009.
- [3] F. Fadilillah, F. Amir, R. Riyanto, R. S. Hasibuan, and S. I. Maryam, “Sistem Informasi Survei Kepuasan Masyarakat dengan Framework Laravel pada Pelayanan Publik Pemerintah Daerah,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 3, pp. 1074–1082, Jul. 2025, doi: 10.57152/malcom.v5i3.1772.
- [4] A. A. Setyogomo, “Penerapan Waterfall dalam Pengembangan Modul Survei Kepuasan Layanan pada Sistem Informasi Akademik Universitas Islam Negeri Antasari Banjarmasin,” *Technologia : Jurnal Ilmiah*, vol. 16, no. 3, pp. 410–417, Jul. 2025, doi: 10.31602/tji.v16i3.18831.
- [5] N. P. Putra and A. Retnowardhani, “Unlocking User Satisfaction: A Delone & Mclean Is Success Model Approach To It Helpdesk Ticketing System Adoption,” *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 6, no. 1, pp. 610–625, Dec. 2024, doi: 10.37385/jaets.v6i1.4469.
- [6] T. V. D. Suyana and N. Legowo, “Analyzing User Satisfaction with the Helpdesk System of PT. Matahari Department Store: An Evaluation Study,” *Journal of Information Systems Engineering and Management*, vol. 10, no. 41s, pp. 1130–1140, May 2025, doi: 10.52783/jisem.v10i41s.8175.
- [7] N. Sharma, “A digital cohort analysis of consumers’ mobile banking app experience,” *Int. J. Consum. Stud.*, vol. 48, no. 1, Jan. 2024, doi: 10.1111/ijcs.12989.
- [8] R. Salama, H. Uzunboylu, and M. A. El Muti, “Implementing online questionnaires and surveys by using mobile applications,” *New Trends and Issues Proceedings on Humanities and Social Sciences*, vol. 7, no. 2, pp. 48–70, Aug. 2020, doi: 10.18844/prosoc.v7i2.5016.

- [9] K. Kishore, S. Khare, V. Uniyal, and S. Verma, "Performance and stability Comparison of React and Flutter: Cross-platform Application Development," in *2022 International Conference on Cyber Resilience (ICCR)*, IEEE, Oct. 2022, pp. 1–4. doi: 10.1109/ICCR56254.2022.9996039.
- [10] F. Pratama and A. Farisi, "Analisis Perbandingan Kinerja Backend API Menggunakan PHP, Golang, dan JavaScript," *Techno.Com*, vol. 24, no. 1, pp. 153–165, Feb. 2025, doi: 10.62411/tc.v24i1.12080.
- [11] Y.-A. Daraghmi, M. Daadoo, and D. Eleyan, "User Centered Method for Enhancing the Adoption of Software Systems in Palestine," in *Proceedings of the International Conference on Future Networks and Distributed Systems*, New York, NY, USA: ACM, Jul. 2017, pp. 1–11. doi: 10.1145/3102304.3102317.
- [12] Y.-A. Daraghmi and E.-Y. Daraghmi, "RAPD: Rapid and Participatory Application Development of Usable Systems During COVID19 Crisis," *IEEE Access*, vol. 10, pp. 93601–93614, 2022, doi: 10.1109/ACCESS.2022.3203582.
- [13] Y. A. Daraghmi, B. Yahya, and Y. Daraghmi, "Has Covid-19 Affected Software Usability: Mobile Accounting System as a Case," *J. Theor. Appl. Inf. Technol.*, vol. 101, no. 2, pp. 785–794, Jan. 2023, Accessed: Dec. 28, 2024. [Online]. Available: <https://www.jatit.org/volumes/Vol101No2/9Vol101No2.pdf>
- [14] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. Ney York: McGraw-Hill Education, 2010.
- [15] Ian. Sommerville, *Software engineering*, 9th ed. Boston: Pearson, 2011.
- [16] R. El-Abidin, "What Is Customer Satisfaction? 5 Reasons It's Important in Service," HubSpot. Accessed: Feb. 15, 2026. [Online]. Available: <https://blog.hubspot.com/service/what-is-customer-satisfaction>
- [17] B. Ives, M. H. Olson, and J. J. Baroudi, "The measurement of user information satisfaction," *Commun. ACM*, vol. 26, no. 10, pp. 785–793, Oct. 1983, doi: 10.1145/358413.358430.
- [18] I. Brace, *How to Plan, Structure and Write Survey Material for Effective Market Research Questionnaire Design*. London: Kogan Page, 2004.
- [19] S. Kumar, "Response Styles of Structured Questions in Business Research," *Asian Review of Social Sciences*, vol. 6, no. 2, pp. 23–28, Nov. 2017, doi: 10.51983/arss-2017.6.2.1371.
- [20] S. Dolnicar and B. Grün, "Does One Size Fit All? The Suitability of Answer Formats for Different Constructs Measured," *Australasian Marketing Journal*, vol. 17, no. 1, pp. 58–64, May 2009, doi: 10.1016/j.ausmj.2009.04.001.

- [21] A. Limited, *ITIL ® Foundation ITIL 4 Edition*, 1st ed. TSO (The Stationery Office), 2019.
- [22] K. C. . Laudon and J. P. . Laudon, *Management Information Systems: Managing the Digital Firm*. United Kingdom: Pearson Education Ltd, 2018.
- [23] A. Hart, “What is Cohort Analysis? Definition, Types and Examples,” Datamation. Accessed: Feb. 15, 2026. [Online]. Available: <https://www.datamation.com/big-data/what-is-cohort-analysis/>
- [24] Samantha. Coyle, *Go programming : from beginner to professional : learn everything you need to build modern software using Go*, 2nd ed. Birmingham: Packt Publishing Ltd., 2024.
- [25] W.-M. Lee, *Beginning Android Application Development*. Indianapolis: Wiley Publishing, Inc., 2011.
- [26] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, “Application Development Using Flutter,” *International Research Journal of Modernization in Engineering Technology and Science @International Research Journal of Modernization in Engineering*, vol. 2, no. 8, pp. 1262–1266, Aug. 2020.
- [27] L. Dagne, “Flutter for cross-platform App and SDK development,” B.E. thesis, Metropolia University of Applied Sciences, Helsinki, 2019.
- [28] P. Luzzanov, E. Rogov, and I. Levshin, *Postgres. The First Experience*, 11th ed. Postgres Professional.
- [29] IONOS, “PostgreSQL: a closer look at the object-relational database management system,” IONOS Digital Guide. Accessed: Feb. 16, 2026. [Online]. Available: <https://www.ionos.ca/digitalguide/server/know-how/postgresql/#c160715>
- [30] A. Kumar S, *Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase*. Birmingham: Packt Publishing, 2018.
- [31] J. Martin, *Rapid application development*. New York: Macmillan Publishing Co., 1991.
- [32] Casemaker Totem Inc, “What is Rapid Application Development?,” 2000. Accessed: Jul. 08, 2024. [Online]. Available: [https://www.iro.umontreal.ca/~dift6803/Transparents/Chapitre1/Documents/rad\\_wp.pdf](https://www.iro.umontreal.ca/~dift6803/Transparents/Chapitre1/Documents/rad_wp.pdf)
- [33] N. B. Hansen *et al.*, “How Participatory Design Works,” in *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, New York, NY, USA: ACM, Dec. 2019, pp. 30–41. doi: 10.1145/3369457.3369460.

- [34] S. Pilemalm and T. Timpka, “Third generation participatory design in health informatics—Making user participation applicable to large-scale information system projects,” *J. Biomed. Inform.*, vol. 41, no. 2, pp. 327–339, Apr. 2008, doi: 10.1016/j.jbi.2007.09.004.
- [35] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual, Second Edition*, 2nd ed. Boston, MA: Addison-Wesley Professional, 2004.
- [36] L. Williams, “Testing Overview and Black-Box Testing Techniques,” in *Testing Overview and Black-Box Testing Techniques*, 2006, pp. 35–59.
- [37] ISTQB, “Certified Tester Foundation Level Syllabus v4.0.1 International Software Testing Qualifications Board,” International Software Testing Qualifications Board, Sep. 2024.
- [38] S. Amland, “Risk-based testing;,” *Journal of Systems and Software*, vol. 53, no. 3, pp. 287–295, Sep. 2000, doi: 10.1016/S0164-1212(00)00019-4.
- [39] R. Firdaus, N. K. Hikmawati, Y. Durachman, H. Nanang, D. Khairani, and M. S. Hazimi, “Usability Testing Analysis of a Company Website in Indonesia,” in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, IEEE, Dec. 2022, pp. 1–6. doi: 10.1109/ICIC56845.2022.10006910.
- [40] N. L. A. K. Y. Sarja, I. P. K. A. Widana, N. K. P. G. Sarja, and I. M. R. A. Nugroho, “Analysis of User Acceptance Learning Management System as Acceleration of Driving School Digitalization,” 2024, pp. 783–792. doi: 10.2991/978-94-6463-622-2\_86.
- [41] M. Cohn, *User Stories Applied for Agile Software Development*. Indiana: Addison Wesley, 2004.
- [42] M. Wynne and A. Hellesøy, *The Cucumber Book: Behaviour-Driven Development for Testers and Developers*. Dallas: Pragmatic Bookshelf, 2012.
- [43] O. Doğan, P. Bozbeyoğlu, and N. Erkoç, “Evaluation of Risks by L-Type (5x5) Matrix Method in an Ore Concentration Structure: A Case Study,” *International Journal of Innovative Approaches in Science Research*, vol. 7, no. 3, pp. 75–90, Oct. 2023, doi: 10.29329/ijiasr.2023.606.2.