

Pengembangan Aplikasi Pengaduan Barang Hilang Berbasis Android dengan Deteksi Penipuan

Daffa Stefian Abyansyah, Masduki Zakarijah
Universitas Negeri Yogyakarta
Email: daffastefian.2020@student.uny.ac.id

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi pengaduan barang hilang berbasis Android yang dilengkapi dengan deteksi penipuan, yang dapat digunakan oleh masyarakat umum untuk membagikan informasi tentang barang hilang atau temuan. Penelitian ini juga bertujuan untuk menilai kelayakan aplikasi tersebut. Metode yang digunakan adalah *Research and Development (R&D)* dengan model pengembangan waterfall, mencakup tahapan *communication, planning, modelling, construction, dan deployment*. Hasil dari penelitian ini adalah aplikasi Nemu yang telah memenuhi standar ISO/IEC 25010. Pada aspek *functional suitability* mendapatkan hasil sebesar 100%. Pada aspek *compatibility* dengan sub karakteristik *co-existence* dan pengujian dengan berbagai perangkat android mendapatkan hasil 100%. Pengujian aspek *performance efficiency* berdasarkan penggunaan CPU hanya sebesar 3.9%, sedangkan penggunaan memori hanya sebesar 2% dan penggunaan energi rata-rata hanya sebesar 126.6 pts. Pada aspek *interaction capability* mendapatkan nilai rata-rata 88.75%.

Kata kunci: android, barang hilang, deteksi penipuan

ABSTRACT

This research aims to design and develop an Android-based lost item complaint application equipped with fraud detection, which can be used by the public to share information about lost or found items. This research also aims to assess the feasibility of the application. The method used is Research and Development (R&D) with a waterfall development model, including the stages of communication, planning, modelling, construction, and deployment. The result of this research is the Nemu application that has met the ISO/IEC 25010 standard. In the aspect of functional suitability, the result is 100%. In the aspect of compatibility with co-existence sub-characteristics and testing with various Android devices, get 100% results. The performance efficiency aspect was tested based on CPU usage, which was only 3.9%, memory usage was only 2%, and average energy usage was only 126.6 pts. In the interaction capability aspect, the average value is 88.75%.

Keywords: Android, lost items, fraud detection

PENDAHULUAN

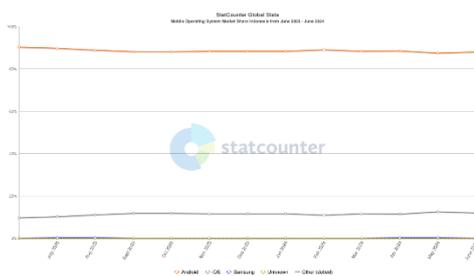
Kehilangan barang merupakan masalah umum yang dialami oleh banyak orang, mulai dari dompet, ponsel, hingga barang berharga lainnya. Selain kerugian material, kehilangan barang juga menimbulkan kekhawatiran dan kecemasan, terutama terkait

penyalahgunaan data kredensial. Dalam Islam, "luqathah" mengacu pada barang hilang yang ditemukan tanpa diketahui pemiliknya. Menurut prinsip Islam, penemu barang hilang harus mengumumkannya kepada masyarakat (Nadzri, 2021).

Perkembangan teknologi di era digital memberikan kemudahan dalam

berbagai aspek, seperti akses informasi, efisiensi kerja, dan komunikasi. Media sosial digunakan untuk berbagi informasi secara cepat dan luas, menjadikannya alternatif platform untuk informasi barang hilang. Namun, informasi di media sosial sering tertimbun dan rawan penipuan. Banyak kasus penipuan terjadi melalui media sosial, seperti modus meminta uang sebagai biaya pengiriman barang hilang. Hal ini meningkatkan potensi aksi kriminal yang merugikan masyarakat.

Teknologi dapat membantu mengatasi masalah ini, salah satunya melalui aplikasi berbasis Android. Android, sebagai sistem operasi *open-source*, memungkinkan pengembangan dan penyesuaian aplikasi (Huda, 2013). Pemanfaatan kecerdasan buatan, khususnya machine learning, dapat menghasilkan prediksi berdasarkan data sebelumnya (Ebert, 2016). Machine learning dapat digunakan untuk mendeteksi dan mencegah penipuan, memberikan keamanan bagi pengguna dalam mencari barang.



Gambar 1. Mobile Operating System Market Share in Indonesia

Berdasarkan Gambar 1., menunjukkan bahwa Android adalah sistem operasi smartphone paling populer di Indonesia, dengan pangsa pasar

87.8% antara Juni 2023 hingga Juni 2024. Pengguna iOS hanya mencapai 12.01%, sementara sisanya menggunakan sistem operasi lain. Kepraktisan smartphone juga menjadikannya perangkat yang penting dalam aktivitas sehari-hari.

Penelitian ini bertujuan mengembangkan aplikasi berbasis Android yang dirancang khusus untuk memfasilitasi pencarian barang hilang atau pemilik barang temuan. Aplikasi ini harus memiliki fitur aman seperti deteksi penipuan dan informasi track record pengguna untuk menghindari aksi kriminal. Dengan adanya platform ini, masyarakat dapat lebih mudah menemukan barang atau pemilik barang yang hilang tanpa khawatir akan penipuan atau informasi yang tertimbun. Serta memastikan produk yang dihasilkan layak sesuai dengan standar (ISO/IEC25010, 2023).

METODE PENELITIAN

Penelitian ini menggunakan metode penelitian *research and development*. Dalam proses pengembangan menggunakan model pengembangan perangkat lunak waterfall menurut (Roger S. Pressman, 2010) yang terdiri dari lima tahapan, yaitu: *communication*, *planning*, *modeling*, *construction*, dan *deployment*. Kemudian dilakukan pengujian menggunakan standar ISO/IEC 25010: 2023 ditinjau dari 4 karakteristik yaitu : yaitu *functional suitability*, *compatibility*, *interaction capability*, dan *performance efficiency*.

Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada bulan Mei 2024 sampai dengan Juli 2024 termasuk dari tahap analisis kebutuhan hingga tahap pendistribusian. Lokasi pengambilan data, pengembangan produk, dan pengujian produk penelitian dilakukan di Universitas Negeri Yogyakarta dan di rumah penulis, yaitu Klaten Jawa Tengah.

Subjek Penelitian

Subjek dalam penelitian ini adalah masyarakat umum yang pernah mengalami kehilangan barang yang ada di wilayah Jogja dan Klaten. Sedangkan objek dari penelitian ini adalah aplikasi pengaduan barang hilang berbasis android dengan fitur deteksi penipuan.

Teknik Pengumpulan Data

Metode yang dilakukan dalam pengumpulan data pada penelitian ini adalah sebagai berikut:

1. Observasi

Metode observasi pada penelitian ini dilakukan dengan mengamati postingan mengenai barang hilang yang seringkali dijumpai pada media sosial. Observasi ini bertujuan untuk mengetahui permasalahan yang ditemui oleh masyarakat yang sedang mencari barang yang hilang dan untuk mengetahui kebutuhan dari sistem yang akan dikembangkan.

2. Kuesioner

Metode kuesioner pada penelitian ini dilakukan dengan membagikan angket kepada para responden yang merupakan

masyarakat umum untuk mengetahui tingkat efektivitas dari aplikasi pengaduan barang hilang berbasis android dengan fitur deteksi penipuan. Angket yang dibagikan adalah pertanyaan mengenai aspek *interaction capability* yang berfungsi untuk menguji sistem. Menurut (Nielsen & Landauer, 1993), jumlah optimal untuk pengujian adalah 7-20 pengguna berdasarkan skala proyek.

Instrumen Penelitian

Instrumen penelitian dijabarkan sesuai dengan instrumen yang digunakan untuk pengujian produk perangkat lunak berdasarkan aspek *functional suitability*, *compatibility*, *interaction capability* dan *performance efficiency*.

1. *Functional Suitability*

Instrumen yang digunakan untuk mengevaluasi ketepatan, kelengkapan, dan kecocokan fungsional aplikasi yang dikembangkan adalah berupa ceklis dengan metode *test case*. Pengujian ini menggunakan skala guttman untuk mendapat jawaban tegas dari suatu permasalahan yang ditanyakan.

2. *Compatibility*

Instrumen yang digunakan untuk menilai kemampuan aplikasi yang dikembangkan untuk berinteraksi dan bertukar informasi dengan sistem lain, serta menjalankan fungsi secara bersamaan adalah berupa ceklis dengan metode *test case* dengan dua sub karakteristik, yaitu *co-existence*: memastikan aplikasi dapat berjalan dengan baik bersamaan dengan aplikasi lain dan instalasi di berbagai perangkat

android: memastikan aplikasi berjalan dengan baik di berbagai perangkat dan sistem operasi yang berbeda. Pengujian ini menggunakan skala guttman.

3. Interaction Capability

Instrumen yang digunakan untuk melakukan pengujian aspek *interaction capability* terhadap hasil penelitian adalah kuesioner yang memiliki 20 pertanyaan dan terbagi dalam 4 kriteria yaitu *ease of use*, *learnability*, *efficiency*, dan *satisfaction* menggunakan 5 skala Likert.

4. Performance Efficiency

Pengujian aplikasi yang dikembangkan berdasarkan aspek *performance efficiency* dilakukan dengan menggunakan Apptim yang merupakan *software* pengujian aplikasi *mobile*. Data yang didapat akan digunakan sebagai indikator dalam sub aspek *time behavior*, *resource utilization*, dan *capacity*.

Teknik Analisis Data

Hasil dari pengujian *functional suitability* dan *compatibility* diperoleh dengan metode *test case* yang dinilai menggunakan skala guttman. Penelitian ini menggunakan opsi "Ya" atau "Tidak". Opsi "Tidak" bernilai 0 dan opsi "Ya" bernilai 1. Berikut rumus untuk perhitungan dari pengujian yang dilakukan pada penelitian ini:

$$Kelayakan(\%) = \frac{\text{nilai total}}{\text{nilai maksimal}} \times 100\%$$

Selanjutnya, hasil pengujian aspek *interaction capability* didapat dengan menggunakan kuesioner yang memiliki 5 pilihan jawaban. Setiap pilihan memiliki skor mulai dari 1 sampai dengan 5.

Adapun pilihan yang tersedia Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), dan Sangat Tidak Setuju (STS). Kemudian dilakukan perhitungan presentase untuk pengujian *interaction capability* dengan menggunakan rumus berikut:

$$Kelayakan(\%) = \frac{\text{nilai total}}{\text{nilai maksimal}} \times 100\%$$

Kemudian nilai presentase yang didapat akan dikonversikan menjadi pernyataan sesuai dengan Tabel 1.

Tabel 1. Interpretasi Kelayakan

No.	Presentase	Kategori
1	0% - 20%	Sangat rendah
2	21% - 40%	Rendah
3	41% - 60%	Cukup
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

Hasil dari pengujian *performance efficiency* diperoleh dengan menggunakan alat pengujian performa aplikasi *mobile* Apptim. Hasil dari pengujian akan digunakan sebagai indikator baik buruknya aspek *performance efficiency*. Tingkat efisiensi aplikasi diukur berdasarkan pada data yang dihasilkan dari Apptim adapun aspek yang diperhatikan dalam pengujian ini diantaranya seperti penggunaan CPU, penggunaan memori, penggunaan jaringan, penggunaan energi, dan *thread*.

HASIL DAN PEMBAHASAN

Pengembangan Aplikasi

a. Communication

Pada penelitian ini, pengembangan produk dimulai dengan tahap menganalisa. Tahap ini dilakukan dengan tujuan untuk mengetahui permasalahan dan kebutuhan masyarakat yang kehilangan barang atau menemukan barang. Metode yang dilakukan adalah observasi, dimana dilakukan pengamatan terhadap kasus-kasus kehilangan barang yang banyak tersebar di sosial media seperti facebook, Instagram, dan whatsapp. Berikut merupakan permasalahan yang dimiliki oleh masyarakat dalam mencari barang hilang yang diperoleh dari tahap observasi:

- 1) Untuk mencari barang yang hilang atau pemilik dari barang temuan biasanya masyarakat membagikan informasi melalui media sosial.
- 2) Informasi kehilangan yang telah dibagikan di media sosial rawan tertimbun karena media sosial merupakan tempat untuk berbagi berbagai macam informasi.
- 3) Tingginya potensi penipuan yang menargetkan masyarakat yang kehilangan barang dengan modus menemukan barang.
- 4) Belum ada aplikasi yang secara khusus dirancang untuk memfasilitasi pencarian barang hilang atau pemilik barang temuan.
- 5) Mayoritas masyarakat menggunakan smartphone berbasis android.

Berdasarkan observasi yang telah dilakukan, dapat diketahui bahwa

masyarakat membutuhkan alat yang dapat memudahkan dalam melakukan pencarian barang hilang atau pemilik barang temuan yang aman.

Setelah kebutuhan dan permasalahan tersebut diketahui, selanjutnya menganalisis kebutuhan yang diperlukan dalam proses pengembangan aplikasi. Adapun hasil dari analisis kebutuhan yang telah dilakukan diantaranya sebagai berikut:

- 1) Analisis Kebutuhan Perangkat Lunak
 - a) Firebase Authentication
 - b) Firebase Realtime Database
 - c) Firebase Storage
 - d) Firebase Dynamic Links
- 2) Analisis Kebutuhan Fungsional
 - a) Pengguna aplikasi harus melakukan autentikasi untuk masuk ke dalam aplikasi.
 - b) Pengguna dapat mendaftarkan diri jika belum memiliki akun.
 - c) Aplikasi dapat digunakan untuk membuat, mengedit, dan menghapus postingan.
 - d) Pengguna dapat melihat informasi yang dibagikan oleh semua pengguna lain dan mencari informasi secara spesifik.
 - e) Pengguna dapat membagikan informasi ke berbagai media lainnya.
 - f) Pengguna dapat berinteraksi dengan pengguna lainnya dengan melakukan percakapan melalui teks.
 - g) Dalam proses interaksi dilengkapi dengan deteksi penipuan dengan menggunakan model klasifikasi teks untuk mencegah aksi kriminalitas.
 - h) Track record pengguna dalam penggunaan aplikasi dapat dilihat

untuk meningkatkan kewaspadaan.

3) Role Pengguna

Sistem aplikasi Nemu dirancang dengan satu role utama yaitu pengguna yang merupakan masyarakat umum yang telah mendaftarkan diri di aplikasi. Pengguna ini dapat mengakses setiap fitur yang terdapat di dalam aplikasi, termasuk membuat, mengedit, dan menghapus postingan, berbagi informasi ke platform lain, serta berinteraksi dengan pengguna lainnya dengan fitur deteksi penipuan untuk menjaga keamanan.

Tabel 2. Perencanaan Timeline Penelitian

No.	Kegiatan	Durasi
1	Analisis masalah	1 Minggu
2	Analisis kebutuhan	2 Minggu
3	Pembuatan UML	2 Minggu
4	Membuat desain antarmuka aplikasi	2 Minggu
5	Pembuatan proyek dan persiapan SDK	1 Hari
6	Pembuatan layout aplikasi	2 Minggu
7	Pembuatan model	2 Minggu
8	Pemrograman aplikasi	4 Minggu
9	Pengujian compatibility	1 Minggu
10	Pengujian performance efficiency	1 Minggu
11	Pengujian functional suitability	1 Minggu
12	Pendistribusian aplikasi	1 Hari
13	Pengujian Interaction Capability	2 Minggu

b. Planning

Perencanaan pengembangan diperlukan agar proses pengembangan

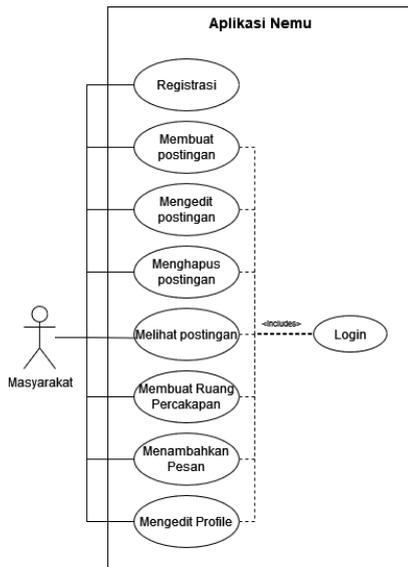
dapat berjalan secara efektif dan sesuai dengan estimasi yang telah diperhitungkan. Pada penelitian ini terdapat beberapa aktivitas yang telah direncanakan waktu pengerjaannya seperti terlihat pada Tabel 2.

Modelling

Pada tahap ini, dilakukan perencanaan dan perancangan mengenai cara kerja dan tampilan aplikasi. Proses perancangan bertujuan untuk menerjemahkan hasil analisis menjadi desain dalam bentuk diagram yang akan diimplementasikan dalam aplikasi. Dalam pengembangan aplikasi Nemu, digunakan dua jenis desain, yaitu desain arsitektur dan desain antarmuka.

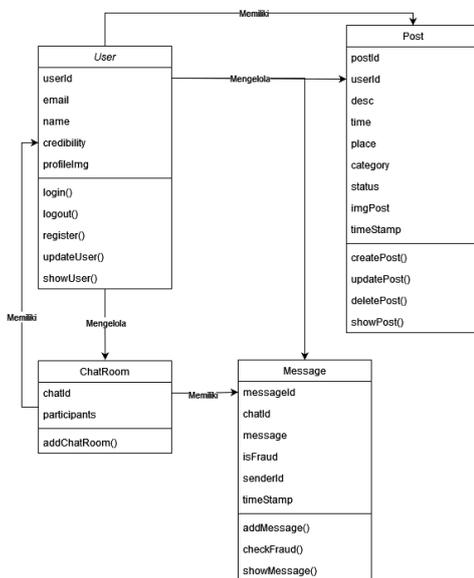
Desain arsitektur aplikasi dibuat dengan menggunakan UML (Unified Modelling Language) yang berfungsi untuk menggambarkan jalannya aplikasi, sehingga dapat memudahkan pengembang untuk mengembangkan produk sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan. Adapun UML yang digunakan adalah *use case diagram*, *sequence diagram*, dan *class diagram*.

Diagram *use case* digunakan untuk menggambarkan hubungan interaksi antara aktor dengan sistem. *Use case* sangat berguna dalam pengembangan perangkat lunak untuk mendefinisikan berbagai interaksi yang terjadi, sebagai gambaran besar sistem yang akan dibangun. Gambar 2. merupakan *use case diagram* dari aplikasi Nemu:



Gambar 2. Use Case Aplikasi Nemu

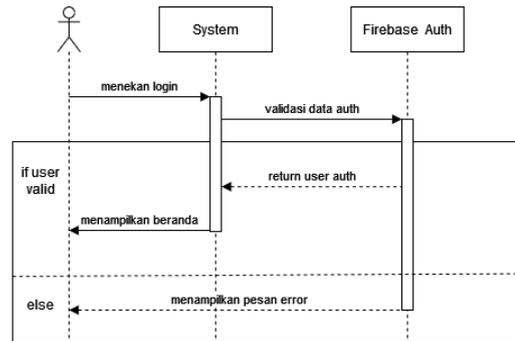
Diagram kelas digunakan untuk merepresentasikan hubungan antar kelas dalam suatu sistem. Diagram ini memudahkan pengembang mendapat gambaran dari struktur dari sistem dengan jelas. Gambar 3. merupakan class diagram dari aplikasi Nemu.



Gambar 3. Class Diagram Aplikasi Nemu

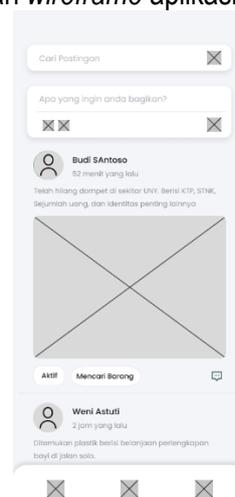
Penggunaan *sequence diagram* bertujuan untuk memperjelas interaksi

antar objek dalam sistem berdasarkan urutan waktu. Diagram ini dapat menggambarkan tahapan yang harus dilalui untuk menghasilkan sesuatu. Gambar 4. merupakan contoh *sequence diagram* yang digunakan untuk menggambarkan proses dari fungsi yang terdapat dalam aplikasi Nemu:



Gambar 4. Sequence Diagram Login

Desain antarmuka adalah rancangan tampilan visual dari aplikasi yang menentukan cara pengguna berinteraksi dengan sistem. Pada proses pengembangan Aplikasi Nemu proses perancangan antarmuka menggunakan *Wireframe* yang merupakan desain visual sederhana dari struktur dan tata letak antarmuka aplikasi. Gambar 5. adalah contoh dari *wireframe* aplikasi Nemu:



Gambar 5. Wireframe Beranda

Construction

Pada tahap develop proses yang dilakukan adalah mengembangkan aplikasi sesuai dengan kebutuhan dan spesifikasi yang telah digambarkan dalam tahap desain. Adapun dalam proses ini dibutuhkan perangkat keras dan perangkat lunak yang diantaranya sebagai berikut:

- a. Perangkat Keras
 - 1) Laptop
 - 2) Smartphone android
- b. Perangkat Lunak
 - 1) Android Studio
 - 2) Firefox
 - 3) Google Colab
 - 4) Draw.io
 - 5) Figma
 - 6) Postman
 - 7) Github

Dalam pengembangan aplikasi Nemu pengembangan dibagi menjadi dua proses, yaitu pengembangan model klasifikasi teks dan pengembangan aplikasi android.

a. Pengembangan Model

1) Pengumpulan Data

Pengumpulan data bertujuan untuk mengumpulkan teks-teks yang berupa pesan-pesan yang dilakukan dalam proses interaksi pencarian barang hilang. Data ini dikumpulkan dari berbagai sosial media, khususnya postingan masyarakat yang menjadi penipuan ketika kehilangan barang.

2) Pelabelan Data

Pelabelan data adalah proses mengidentifikasi dan mengklasifi-

kasikan data mentah ke dalam kelas tertentu agar model dapat mempelajari dan memprediksi data baru yang belum diberi label. Dalam penelitian ini, data diklasifikasikan menjadi dua kelas: *fraud* dan *non-fraud*. Proses pelabelan dilakukan dengan mengidentifikasi karakteristik masing-masing kelas untuk memastikan data yang dilabeli merepresentasikan kelas tersebut. Observasi menunjukkan karakteristik pesan penipuan meliputi: pengirim mengaku berada di luar kota, menyarankan pengiriman barang melalui ekspedisi, dan meminta korban mentransfer biaya pengiriman terlebih dahulu.

3) Pembuatan Model

Dalam membuat model, data yang telah diklasifikasikan perlu diolah terlebih dahulu. Untuk data teks, tahap *pre process* mencakup penghapusan angka, karakter, *stop words*, dan pengubahan kata ke bentuk dasar. Setelah data dibersihkan dan dinormalisasi, data diekstrak menjadi fitur numerik agar dapat diolah oleh model. Gambar 6. adalah kode untuk menjalankan proses tersebut.

```

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'\W+', ' ', text)
    text = re.sub(r'\s+', ' ', text)
    text = text.strip()

    stop_words = set(stopwords.words('Indonesian'))
    lemmatizer = WordNetLemmatizer()

    words = text.split()
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]

    return ' '.join(words)

df = pd.DataFrame(data)

df['text'] = [preprocess_text(text) for text in df['text']]

vectorizer = CountVecorizer()
x = vectorizer.fit_transform(df['text'])
y = df['label']

model_klasifikasi = MultinomialNB()
model_klasifikasi.fit(x, y)

```

Gambar 6. Kode Pembuatan Model

4) Evaluasi Model

Untuk memastikan model memiliki tingkat akurasi tinggi dan dapat menghasilkan prediksi yang tepat, evaluasi dilakukan menggunakan data uji berlabel yang terdiri dari 10 kalimat. Prediksi dari model yang telah dilatih kemudian dibandingkan dengan label asli dari data uji. Gambar 7. adalah hasil evaluasi yang telah dilakukan.

Evaluation Accuracy: 0.8				
Evaluation Classification Report:				
	precision	recall	f1-score	support
fraud	1.00	0.60	0.75	5
non_fraud	0.71	1.00	0.83	5
accuracy			0.80	10
macro avg	0.86	0.80	0.79	10
weighted avg	0.86	0.80	0.79	10

Gambar 7. Hasil Evaluasi Model

b. Pengembangan Aplikasi Android

1) Pembuatan *Layout* Aplikasi

Layout menentukan tampilan dan interaksi pengguna dalam aplikasi. Dalam pengembangan aplikasi Nemu, *layout* dibangun menggunakan XML (Extensible Markup Language) sesuai dengan desain *wireframe* yang telah dibuat, sehingga memudahkan

pengembang dalam mendefinisikan komponen.

XML yang didefinisikan akan di-*render* oleh aplikasi untuk menghasilkan antarmuka pengguna. Proses pembuatan *layout* dilakukan di Android Studio, yang memungkinkan pengembang melihat hasil tampilan secara langsung. Gambar 8. adalah *layout* yang diimplementasikan:



Gambar 8. *Layout* Beranda

2) Pemrograman Aplikasi

Setelah *layout* selesai, perlu dilakukan pemrograman agar tampilan dapat berinteraksi dengan pengguna dan menjalankan fungsi yang dibutuhkan. Aplikasi ini menerapkan arsitektur MVVM untuk memudahkan pengembangan yang efisien dan terstruktur. Gambar 9. menunjukkan implementasi pemrograman aplikasi Nemu:

b. Hasil Pengujian *Compatibility*

Pengujian aspek *compatibility* terhadap aplikasi Nemu dilakukan dengan melakukan dua jenis pengujian yaitu pengujian *co-existence* dan pengujian pada berbagai perangkat android.

Pengujian sub karakteristik *co-existence* dilakukan dengan menjalankan aplikasi Nemu secara bersamaan dengan 10 aplikasi yang populer di masyarakat. Pengujian ini mendapatkan skor 10 dari 10 pengujian, sehingga hasil dari pengujian *compatibility* berdasarkan karakteristik *co-existence* memperoleh hasil presentase kelayakan sebagai berikut:

$$\begin{aligned} kelayakan &= \frac{\text{nilai total}}{\text{nilai maksimal}} \times 100\% \\ &= \frac{10}{10} \times 100\% = 100\% \end{aligned}$$

Hasil dari pengujian *co-existence* mendapatkan hasil sebesar 100% yang berarti bahwa aplikasi Nemu dapat berfungsi dengan baik bersamaan dengan aplikasi lainnya tanpa ada kendala. Sehingga dapat disimpulkan dari segi *co-existence* aplikasi Nemu dikategorikan dalam "Sangat Baik".

Pengujian *compatibility* dengan sub karakteristik berikutnya dilakukan dengan menginstal aplikasi Nemu pada beberapa perangkat android untuk memastikan aplikasi dapat berfungsi dengan baik pada berbagai perangkat android masyarakat yang beraneka ragam. Pengujian instalasi mendapatkan hasil lolos 10 dari 10 pengujian. Sehingga hasil dari pengujian *compatibility*

berdasarkan instalasi ke berbagai perangkat android memperoleh hasil presentase kelayakan sebagai berikut:

$$\begin{aligned} kelayakan &= \frac{\text{nilai total}}{\text{nilai maksimal}} \times 100\% \\ &= \frac{10}{10} \times 100\% = 100\% \end{aligned}$$

Hasil presentase kelayakan menunjukkan nilai 100%. Hal tersebut berarti bahwa aplikasi Nemu dapat berjalan dengan baik di berbagai perangkat yang memiliki versi android berbeda. Sehingga dapat disimpulkan dari sub karakteristik pengujian dengan berbagai versi android, aplikasi Nemu dikategorikan dalam "Sangat Baik".

c. Hasil Pengujian Performance Efficiency

Pengujian aspek *performance efficiency* dari aplikasi Nemu dilakukan dengan menggunakan perangkat android, adapun skema yang dilakukan dalam pengujian ini ditunjukkan pada Tabel 4.

Tabel 4. Daftar Aktivitas Skema Pengujian

No.	Aktivitas
1	Masuk dengan google
2	Melihat postingan
3	Mengirim pesan
4	Membuat postingan
5	Melihat profil pengguna
6	Keluar dari akun

Setelah melakukan aktivitas yang terdapat pada tabel 4, maka dapat dilihat hasil pengujian aspek *performance efficiency*.

Penggunaan CPU aplikasi Nemu rata-rata hanya sebesar 3.9%. Berdasarkan nilai yang didapat, maka aplikasi nemu dapat dikatakan efisien dalam penggunaan CPU.

Penggunaan memori mendapatkan nilai rata-rata sebesar 167.1 MB. Hal tersebut berarti aplikasi rata-rata hanya menggunakan 2% dari total memori yang dimiliki oleh perangkat pengujian. Maka dari itu aplikasi Nemu dapat dikatakan efisien memori.

Berdasarkan aspek penggunaan energi, Aplikasi nemu mendapatkan angka penggunaan energi rata-rata sebesar 126.6 pts. Ambang batas dari aplikasi dapat dikatakan ringan adalah 250 pts. Sehingga aplikasi Nemu dapat dikatakan efisien energi.

Pada pengujian penggunaan jaringan, besaran penggunaan jaringan aplikasi Nemu dalam skema pengujian yang telah dijalankan, menunjukkan total *download* sebesar 0.9 MB dan total *upload* sebesar 0.3 MB. Sehingga aplikasi Nemu dapat dikatakan efisien dalam penggunaan jaringan.

Hasil pengujian aspek *time behavior*, aplikasi Nemu rata-rata menghasilkan rata-rata 68 dengan nilai *threads* tertinggi 73 *threads/s* dan nilai *threads* terendah 48 *threads/s*. Dalam mengeksekusi *thread* dengan jumlah tersebut aplikasi tidak mengalami *crash* dan penggunaan CPU juga sangat kecil.

d. Hasil Pengujian *Interaction Capability*

Pengujian aspek *interaction capability* terhadap aplikasi Nemu dilakukan dengan menggunakan kuesioner dengan 20 pertanyaan. Kuesioner dibagikan kepada masyarakat umum yang merupakan telah menggunakan aplikasi Nemu. Tabel 5. merupakan hasil dari pengujian *interaction capability*.

Tabel 5. Hasil Pengujian *Interaction Capability*

No	Nama	Jumlah	Presentase
1	Bagas	88	88
2	Siti	82	82
3	Alex	90	90
4	Santa	83	83
5	Dela	85	85
6	Via	95	95
7	Wahyu	86	86
8	Widakdo	97	97
9	Bagus	81	81
10	Fadhilah	90	90
11	Daffa	88	88
12	Sholeh	82	82
13	Widya	97	97
14	Silvia	98	98
15	Rara	98	98
16	Hilmy	71	71
17	Mario	95	95
18	Hannan	92	92
19	Ishaq	90	90
20	Kurniawan	87	87
Rata-rata			88.75

Hasil pengujian *interaction capability* pada tabel 5, menunjukkan nilai rata-rata dari pengujian aspek *interaction capability* terhadap aplikasi Nemu mendapatkan nilai 88,75%. Dengan demikian, dapat disimpulkan bahwa dari aspek *interaction capability*, aplikasi Nemu dikategorikan dalam "Sangat Baik".

KESIMPULAN

Berdasarkan hasil dan pembahasan penelitian di atas, maka dapat diambil bahwa Aplikasi Nemu dikembangkan menggunakan model *waterfall* melalui tahap *communication*, *planning*, *modelling*, *construction*, dan *deployment*. Pengembangan meliputi model deteksi penipuan di Google Colab dengan Python dan aplikasi Android di Android Studio menggunakan Kotlin serta layanan Firebase sebagai backend. Aplikasi Nemu berhasil menyediakan fitur berbagi informasi, postingan ke platform lain, percakapan, deteksi pesan penipuan, dan informasi track record pengguna, sehingga menjawab kebutuhan masyarakat dalam pengaduan barang hilang dengan aman.

Pengujian kelayakan aplikasi Nemu menggunakan standar ISO/IEC 25010 menunjukkan kualitas yang sangat baik. Pada aspek *functional suitability*, aplikasi dinilai sangat baik dengan 100%. Dalam aspek *compatibility*, aplikasi mendapat 100% untuk sub-karakteristik *co-existence* dan kompatibilitas dengan berbagai perangkat Android. Pada aspek *performance efficiency*, Nemu dinilai sangat baik dalam penggunaan CPU (3.9%), memori (2%), energi (126.6 pts),

dan jaringan. Aspek *interaction capability* mendapat penilaian 88.75%. Secara keseluruhan, aplikasi Nemu dinyatakan sangat baik dan layak digunakan untuk berbagi informasi barang hilang atau temuan dengan aman.

DAFTAR RUJUKAN

- Ebert, P. L. C. (2016). Machine Learning. *IEEE Software*, 33, 110–115.
- Huda, A. A. (2013). *Live coding! 9 aplikasi android buatan sendiri* (1st ed.). ANDI.
- ISO/IEC25010. (2023). Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Product quality model. *Switzerland*, 2023.
- Nadzri, N. I. B. M. (2021). *Hukum Mengambil Barang Temuan (Luqathah)(Studi Komparatif Antara Imam Malik Dan Imam Syafi'i)*. 76.
- Nielsen, J., & Landauer, T. K. (1993). Mathematical model of the finding of usability problems. *Conference on Human Factors in Computing Systems - Proceedings*, 206–213. <https://doi.org/10.1145/169059.169166>
- Roger S. Pressman. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). McGraw-Hill Education.