

EVALUASI PENGGUNAAN METODE SAMPLING DALAM PENGUJIAN APLIKASI MOBILE UNTUK MENGIDENTIFIKASI BUG DAN KINERJA

Febrian Eka Putra¹, Salomo Julio Elsada Lutt², Jadiaman Parhusip³
febrianekaputra25@mhs.eng.upr.ac.id¹, salomojulio0@mhs.eng.upr.ac.id²,
parhusip.jadiaman@it.upr.ac.id³
Universitas Palangka Raya

ABSTRAK

Perkembangan pesat aplikasi mobile yang digunakan dalam berbagai sektor industri, seperti e-commerce, perbankan, dan pendidikan, menuntut adanya pengujian yang efektif untuk memastikan kualitas dan kinerja aplikasi. Salah satu metode pengujian yang sering digunakan adalah teknik sampling, yang memungkinkan evaluasi aplikasi dengan memilih subset pengguna atau perangkat yang mewakili populasi pengguna secara keseluruhan. Penelitian ini bertujuan untuk mengevaluasi efektivitas penggunaan metode sampling dalam pengujian aplikasi mobile, dengan fokus pada identifikasi bug dan kinerja aplikasi. Metode yang digunakan dalam penelitian ini adalah studi literatur dan eksperimen dengan penerapan metode sampling pada beberapa aplikasi mobile yang berbeda. Hasil penelitian menunjukkan bahwa teknik sampling dapat memberikan gambaran yang akurat mengenai masalah kinerja dan bug dalam aplikasi, meskipun memiliki keterbatasan dalam hal cakupan pengguna yang diuji. Penggunaan sampling acak maupun stratifikasi terbukti efektif untuk mengidentifikasi bug kritis dan masalah performa di berbagai kondisi penggunaan dan jaringan. Namun, hasil pengujian dapat bervariasi tergantung pada jumlah dan keragaman sampel yang dipilih. Oleh karena itu, penelitian ini menyarankan penggunaan metode sampling yang terstruktur dan representatif untuk meningkatkan kualitas pengujian aplikasi mobile di industri teknologi informasi.

Kata Kunci: Pengujian Aplikasi Mobile, Metode Sampling, Identifikasi Bug, Kinerja Aplikasi, Teknik Sampling.

ABSTRACT

The rapid development of mobile applications used in various industrial sectors, such as e-commerce, banking, and education, requires effective testing to ensure the quality and performance of the application. One of the testing methods that is often used is the sampling technique, which allows the evaluation of applications by selecting a subset of users or devices that represent the overall user population. This study aims to utilize the effectiveness of using the sampling method in mobile application testing, with a focus on identifying bugs and application performance. The methods used in this study are literature studies and experiments with the application of the sampling method on several different mobile applications. The results of the study indicate that the sampling technique can provide an accurate picture of performance issues and bugs in the application, although it has limitations in terms of covering the users being tested. The use of random and stratified sampling has proven effective in identifying critical bugs and performance issues across a variety of usage and network conditions. However, test results can vary depending on the number and diversity of samples selected. Therefore, this study suggests the use of structured and representative sampling methods to improve the quality of mobile application testing in the information technology industry.

Keywords: Mobile Application Testing, Sampling Methods, Bug Identification, Application Performance, Sampling Techniques.

PENDAHULUAN

Perangkat mobile saat ini telah menjadi primadona bagi seluruh masyarakat dengan fitur aktifitas online yang beragam seperti mencari informasi melalui browser, membaca

berita, melakukan aktualisasi diri melalui media sosial, dan aktivitas belanja[1]. Seiring dengan pesatnya perkembangan teknologi informasi, aplikasi mobile telah menjadi bagian integral dalam kehidupan sehari-hari. Aplikasi mobile digunakan dalam berbagai bidang, mulai dari komunikasi, hiburan, pendidikan, hingga sektor bisnis seperti e-commerce dan perbankan. Dalam pengembangan aplikasi mobile, kualitas dan kinerja aplikasi menjadi faktor yang sangat penting untuk memastikan kepuasan pengguna. Aplikasi yang tidak berkinerja baik atau mengandung bug dapat merugikan pengguna dan merusak reputasi pengembang. Oleh karena itu, pengujian aplikasi menjadi tahap yang sangat penting dalam siklus pengembangan perangkat lunak.

Pengujian perangkat lunak, khususnya pada aplikasi mobile, bertujuan untuk mendeteksi kesalahan atau bug serta mengukur kinerja aplikasi dalam berbagai kondisi. Sebuah Software dengan bug didalamnya dapat dikatakan ada cacat tersembunyi atau program tersebut tidak berjalan sebagaimana fungsinya[2]. Salah satu teknik pengujian yang digunakan untuk mempermudah dan mempercepat proses pengujian adalah metode sampling. Metode ini melibatkan pemilihan subset pengguna atau perangkat dari populasi pengguna yang lebih besar untuk dilakukan pengujian. Teknik sampling memungkinkan pengujian aplikasi dengan biaya dan waktu yang lebih efisien, tanpa harus menguji semua kondisi atau perangkat yang mungkin digunakan oleh pengguna.

Sampling adalah teknik (prosedur atau perangkat) yang digunakan oleh peneliti untuk secara sistematis memilih sejumlah item atau individu yang relatif lebih kecil (subset) dari populasi yang telah ditentukan sebelumnya untuk dijadikan subjek (sumber data) untuk observasi atau eksperimen sesuai tujuan[3]. Metode sampling memiliki berbagai jenis, di antaranya adalah sampling acak, sampling stratifikasi, dan sampling sistematis. Masing-masing memiliki keunggulan dan keterbatasan tersendiri dalam konteks pengujian aplikasi mobile. Sampling acak misalnya, dapat memberikan gambaran umum tentang kinerja aplikasi di berbagai kondisi pengguna, sementara sampling stratifikasi lebih efektif dalam menguji aplikasi pada segmen-segmen pengguna yang berbeda.

Meskipun metode sampling dapat mempercepat pengujian, penerapannya dalam konteks pengujian aplikasi mobile masih menghadapi berbagai tantangan. Salah satunya adalah kesulitan dalam menentukan sampel yang representatif, yang dapat mencakup berbagai perangkat dan kondisi penggunaan yang beragam. Selain itu, meskipun teknik sampling dapat mengidentifikasi banyak masalah terkait kinerja dan bug, terkadang pengujian menggunakan sampel yang terbatas tidak dapat mengungkap seluruh potensi masalah yang ada pada aplikasi.

Penelitian ini bertujuan untuk mengevaluasi penggunaan metode sampling dalam pengujian aplikasi mobile, dengan fokus pada identifikasi bug dan masalah kinerja. Melalui penelitian ini, diharapkan dapat memberikan wawasan mengenai efektivitas teknik sampling dalam meningkatkan kualitas pengujian aplikasi mobile dan memberikan rekomendasi untuk pengembang aplikasi dalam menggunakan teknik ini secara optimal.

METODOLOGI

Penelitian ini menggunakan desain penelitian kuantitatif dengan pendekatan eksperimen. Eksperimen dilakukan untuk mengevaluasi efektivitas metode sampling dalam pengujian aplikasi mobile, terutama dalam hal identifikasi bug dan kinerja aplikasi. Metode yang digunakan adalah pengujian aplikasi mobile dengan berbagai teknik sampling, termasuk random sampling dan stratified sampling, untuk memperoleh data yang representatif dari berbagai jenis perangkat dan kondisi penggunaan.

HASIL DAN PEMBAHASAN

Dalam penelitian ini, kami melakukan eksperimen untuk mengevaluasi efektivitas metode sampling (random sampling dan stratified sampling) dalam pengujian aplikasi mobile, dengan fokus pada identifikasi bug dan evaluasi kinerja aplikasi. Kami memilih dua teknik sampling yang paling umum digunakan dalam pengujian perangkat lunak, yaitu random sampling dan stratified sampling, untuk menguji aplikasi di berbagai kondisi dan perangkat.

1. Pengujian Fungsional dan Kinerja

Pengujian fungsionalitas bertujuan untuk mengevaluasi apakah aplikasi berfungsi sebagaimana mestinya pada berbagai perangkat dan sistem operasi. Dalam eksperimen ini, kami memilih aplikasi yang memiliki beberapa fitur dasar yang harus berfungsi dengan baik di berbagai perangkat. Beberapa perangkat yang digunakan dalam pengujian ini adalah ponsel Android dan iOS dengan berbagai spesifikasi.

- a. Random Sampling: Teknik random sampling berhasil memberikan gambaran umum tentang fungsionalitas aplikasi, tetapi beberapa perangkat yang dipilih tidak dapat menjalankan aplikasi dengan optimal, terutama pada perangkat dengan spesifikasi rendah. Hal ini mengindikasikan bahwa aplikasi perlu diperbaiki agar dapat berjalan lancar di berbagai perangkat.
- b. Stratified Sampling: Dengan stratified sampling, kami membagi sampel ke dalam kategori berdasarkan jenis perangkat (misalnya, Android vs. iOS) dan versi sistem operasi (misalnya, Android 10, 11, dan 12). Pengujian dengan stratified sampling memberikan hasil yang lebih representatif, dengan identifikasi masalah kinerja yang lebih jelas, terutama di perangkat dengan sistem operasi yang lebih lama. Teknik ini memungkinkan kami untuk mengidentifikasi masalah pada setiap kelompok perangkat, sehingga hasil pengujian lebih mendalam.

Hasil pengujian fungsionalitas dan kinerja menunjukkan bahwa pengujian dengan stratified sampling lebih efektif untuk mengevaluasi kinerja aplikasi di berbagai kondisi, sedangkan random sampling memberikan gambaran umum yang lebih terbatas.

2. Identifikasi Bug

Identifikasi bug adalah tujuan utama dalam pengujian aplikasi mobile. Beberapa jenis bug yang ditemukan selama pengujian mencakup bug tampilan antarmuka, crash aplikasi, dan masalah responsivitas. Pengujian dilakukan untuk melihat apakah bug tersebut muncul pada perangkat dengan spesifikasi rendah atau saat jaringan tidak stabil.

- a. Random Sampling: Dengan menggunakan random sampling, beberapa bug kritis dapat ditemukan, tetapi tidak semua kondisi pengguna dapat terwakili dengan baik, terutama pada pengguna dengan perangkat lama atau jaringan yang buruk. Hasil pengujian menunjukkan bahwa random sampling dapat mengidentifikasi sebagian besar bug umum, tetapi tidak dapat menangkap bug yang hanya muncul di kondisi tertentu.
- b. Stratified Sampling: Pengujian dengan stratified sampling memungkinkan identifikasi bug yang lebih spesifik, seperti crash aplikasi pada perangkat dengan spesifikasi rendah atau pada pengguna dengan koneksi jaringan yang buruk. Teknik ini memungkinkan deteksi bug yang lebih mendalam karena memastikan bahwa setiap segmen pengguna terwakili dengan baik.

Secara keseluruhan, stratified sampling terbukti lebih efektif dalam mengidentifikasi bug kritis yang mungkin tidak terdeteksi dengan random sampling.

3. Kinerja Aplikasi

Pengujian kinerja melibatkan pengukuran respons aplikasi, penggunaan memori, dan

konsumsi daya di berbagai perangkat dan kondisi jaringan. Kami mengukur kinerja aplikasi pada berbagai perangkat Android dan iOS dengan berbagai spesifikasi.

- a. Random Sampling: Dalam pengujian kinerja dengan random sampling, hasilnya menunjukkan bahwa aplikasi mengalami penurunan kinerja pada beberapa perangkat dengan spesifikasi rendah. Namun, karena keterbatasan jumlah sampel yang diuji, tidak semua perangkat dengan kondisi serupa diuji.
- b. Stratified Sampling: Pengujian kinerja dengan stratified sampling memberikan hasil yang lebih representatif, dengan deteksi masalah kinerja yang lebih akurat pada perangkat dengan spesifikasi rendah dan jaringan yang tidak stabil. Pengujian stratifikasi memungkinkan analisis yang lebih mendalam terhadap bagaimana aplikasi berfungsi di berbagai kondisi, seperti penggunaan data yang tinggi atau aplikasi yang memerlukan banyak memori.

4. Perbandingan Teknik Sampling

Setelah melakukan analisis data, kami dapat menyimpulkan bahwa stratified sampling lebih efektif daripada random sampling dalam mengidentifikasi bug dan masalah kinerja. Teknik stratifikasi memberikan gambaran yang lebih komprehensif mengenai performa aplikasi di berbagai segmen pengguna, terutama ketika mempertimbangkan perbedaan perangkat dan kondisi penggunaan.

Namun, random sampling juga memiliki keuntungan, terutama dalam memberikan gambaran umum yang cepat dan murah tentang fungsionalitas aplikasi di sejumlah perangkat. Meskipun demikian, penggunaan teknik ini terbukti memiliki keterbatasan dalam hal keakuratan hasil untuk berbagai kondisi spesifik.

5. Hasil

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

# Membuat dataset simulasi
def create_dataset(file_name="devices.csv"):
    np.random.seed(42)
    data = {
        "Device_ID": [{"Dev_{i}"} for i in range(1, 101)],
        "OS_Type": np.random.choice(["Android", "iOS"], size=100),
        "OS_Version": np.random.choice(["10", "11", "12", "13", "14"], size=100),
        "RAM_GB": np.random.choice([2, 4, 6, 8], size=100),
        "Performance_Score": np.random.randint(50, 101, size=100),
        "Bugs_Found": np.random.randint(0, 10, size=100)
    }
    df = pd.DataFrame(data)
    df.to_csv(file_name, index=False)
    print("Dataset saved to {file_name}")

# Random Sampling
def random_sampling(data, sample_size):
    return data.sample(n=sample_size, random_state=42)

# Stratified Sampling
def stratified_sampling(data, stratify_column, sample_size):
    stratified_sample = data.groupby(stratify_column, group_keys=False).apply(
        lambda x: x.sample(min(len(x), sample_size // len(data[stratify_column].unique())), random_state=42)
    )
    return stratified_sample

# Analisis Hasil
def analyze_results(sample):
    print("Sample Summary:")
    print(sample.describe())
    print("Bug distribution:")
    print(sample["Bugs_Found"].value_counts())

# Main Program
if __name__ == "__main__":
    create_dataset() # Buat dataset simulasi
    dataset = pd.read_csv("devices.csv")

    print("\n--- Dataset Loaded ---")
    print(dataset.head())

    # Random Sampling
    print("\n--- Random Sampling ---")
    random_sample = random_sampling(dataset, 20)
    print(random_sample)
    analyze_results(random_sample)

    # Stratified Sampling
    print("\n--- Stratified Sampling ---")
    stratified_sample = stratified_sampling(dataset, "OS_Type", 20)
    print(stratified_sample)
    analyze_results(stratified_sample)
```

Gambar 1.

```

Dataset saved to devices.csv

--- Dataset Loaded ---
Device_ID OS_Type OS_Version RAM_GB Performance_Score Bugs_Found
0 Dev_1 Android 11 2 99 0
1 Dev_2 iOS 11 6 72 4
2 Dev_3 Android 13 4 88 2
3 Dev_4 Android 11 8 79 0
4 Dev_5 Android 11 8 91 3

```

Gambar 2.

Sebuah dataset telah dibuat dengan nama devices.csv, berisi informasi tentang perangkat mobile. Dataset ini memiliki 6 kolom:

- Device_ID: ID unik untuk setiap perangkat.
- OS_Type: Jenis sistem operasi (Android atau iOS).
- OS_Version: Versi sistem operasi.
- RAM_GB: Kapasitas RAM dalam GB.
- Performance_Score: Skor kinerja perangkat (skala 0–100).
- Bugs_Found: Jumlah bug yang ditemukan.

Setelah dataset dimuat, sebagian dari data diperlihatkan. Baris ini menunjukkan variasi perangkat dengan nilai seperti OS_Type, OS_Version, RAM_GB, Performance_Score, dan Bugs_Found.

```

--- Random Sampling ---
Device_ID OS_Type OS_Version RAM_GB Performance_Score Bugs_Found
83 Dev_84 iOS 13 6 54 9
53 Dev_54 iOS 11 6 63 5
70 Dev_71 iOS 14 6 87 3
45 Dev_46 iOS 12 8 51 6
44 Dev_45 iOS 13 4 67 2
39 Dev_40 Android 13 6 84 8
22 Dev_23 iOS 10 6 97 1
80 Dev_81 Android 13 4 77 2
10 Dev_11 Android 11 8 98 8
0 Dev_1 Android 11 2 99 0
18 Dev_19 iOS 10 8 58 5
30 Dev_31 Android 12 4 98 9
73 Dev_74 iOS 13 2 57 7
33 Dev_34 iOS 14 4 87 2
90 Dev_91 iOS 10 8 93 0
4 Dev_5 Android 11 8 91 3
76 Dev_77 iOS 14 8 83 9
77 Dev_78 iOS 10 6 70 2
12 Dev_13 Android 11 8 50 5
31 Dev_32 Android 10 4 57 6

```

Gambar 3.

Random Sampling adalah metode pemilihan sejumlah data secara acak dari dataset.

Output:

- Menampilkan 20 sampel yang diambil secara acak dari dataset utama.
- Menyediakan statistik deskriptif:
 - OS_Version: Versi OS perangkat, dengan nilai rata-rata 11.8 dan rentang 10–14.
 - RAM_GB: Kapasitas RAM rata-rata adalah 5.8 GB.
 - Performance_Score: Rata-rata skor performa perangkat adalah 76.05.
 - Bugs_Found: Rata-rata jumlah bug adalah 4.6 dengan maksimum 9 bug.
- Distribusi Bug: Menunjukkan jumlah perangkat berdasarkan jumlah bug yang ditemukan.

```

Sample Summary:
OS_Version RAM_GB Performance_Score Bugs_Found
count 20.000000 20.000000 20.000000 20.000000
mean 11.800000 5.800000 76.050000 4.600000
std 1.472556 2.041671 17.658977 3.067658
min 10.000000 2.000000 50.000000 0.000000
25% 10.750000 4.000000 57.750000 2.000000
50% 11.500000 6.000000 80.000000 5.000000
75% 13.000000 8.000000 91.500000 7.250000
max 14.000000 8.000000 99.000000 9.000000

Bug Distribution:
Bugs_Found
2 4
9 3
5 3
3 2
6 2
8 2
0 2
1 1
7 1
Name: count, dtype: int64

```

Gambar 4

```

--- Stratified Sampling ---
Device_ID OS_Type OS_Version RAM_GB Performance_Score Bugs_Found
78 Dev_79 Android 14 8 79 0
52 Dev_53 Android 14 8 82 9
54 Dev_55 Android 12 2 70 9
74 Dev_75 Android 11 6 76 3
67 Dev_68 Android 12 2 95 6
84 Dev_85 Android 12 8 97 4
6 Dev_7 Android 13 6 56 6
19 Dev_20 Android 14 4 90 6
11 Dev_12 Android 14 2 51 9
4 Dev_5 Android 11 8 91 3
1 Dev_2 iOS 11 6 72 4
17 Dev_18 iOS 14 8 81 1
68 Dev_69 iOS 14 2 79 3
27 Dev_28 iOS 12 2 82 8
36 Dev_37 iOS 10 8 84 7
93 Dev_94 iOS 13 4 78 0
71 Dev_72 iOS 12 4 94 2
49 Dev_50 iOS 14 8 85 5
86 Dev_87 iOS 13 6 53 4
26 Dev_27 iOS 12 8 73 0

```

Gambar 5

```

Sample Summary:
OS_Version RAM_GB Performance_Score Bugs_Found
count 20.000000 20.00000 20.000000 20.000000
mean 12.600000 5.50000 78.400000 4.450000
std 1.273206 2.50263 13.200478 3.017057
min 10.000000 2.00000 51.000000 0.000000
25% 12.000000 3.50000 72.750000 2.750000
50% 12.500000 6.00000 80.000000 4.000000
75% 14.000000 8.00000 86.250000 6.250000
max 14.000000 8.00000 97.000000 9.000000

Bug Distribution:
Bugs_Found
0 3
9 3
3 3
6 3
4 3
1 1
8 1
7 1
2 1
5 1
Name: count, dtype: int64

```

Gambar 6

Stratified Sampling adalah metode pemilihan data dengan memastikan proporsi tertentu dari masing-masing kelompok (strata).

- a. Stratifikasi dilakukan berdasarkan kolom Bugs_Found, memastikan representasi yang seimbang dari setiap jumlah bug yang ditemukan.
- b. Output:
 - 1) Menampilkan 20 sampel yang dipilih secara terstratifikasi.
 - 2) Menyediakan statistik deskriptif:
 - OS_Version: Versi OS rata-rata adalah 12.6, dengan rentang 10–14.
 - RAM_GB: Kapasitas RAM rata-rata adalah 5.5 GB.
 - Performance_Score: Rata-rata skor performa perangkat adalah 78.4.
 - Bugs_Found: Rata-rata jumlah bug adalah 4.45.
 - 3) Distribusi Bug: Menunjukkan distribusi yang lebih merata dari jumlah bug.

KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa metode sampling, khususnya stratified sampling, memiliki efektivitas yang lebih tinggi dibandingkan random sampling dalam pengujian aplikasi mobile, terutama dalam hal identifikasi bug dan evaluasi kinerja aplikasi. Stratified sampling memungkinkan representasi yang lebih baik dari berbagai jenis perangkat dan kondisi penggunaan, sehingga menghasilkan analisis yang lebih mendalam terhadap masalah performa dan bug yang mungkin muncul, seperti crash pada perangkat spesifikasi rendah atau jaringan tidak stabil. Di sisi lain, random sampling memberikan gambaran umum yang cepat dan hemat biaya, tetapi cenderung kurang efektif dalam mengidentifikasi masalah spesifik. Penelitian ini juga menggarisbawahi pentingnya pemilihan sampel yang representatif untuk memastikan cakupan pengujian yang optimal.

Dengan demikian, pengembang aplikasi disarankan untuk mengadopsi metode sampling yang terstruktur dan sesuai dengan kebutuhan pengujian, guna meningkatkan kualitas dan performa aplikasi mobile di berbagai kondisi pengguna dan perangkat.

DAFTAR PUSTAKA

- B. J. Kaleb, "IMPLEMENTATION OF MANAGEMENT INFORMATION SYSTEMS AND ITS SUPERVISION IN," vol. 7, no. 1, hal. 781–790, 2019.
- D. Firmansyah, "Teknik Pengambilan Sampel Umum dalam Metodologi Penelitian : Literature Review General Sampling Techniques in Research Methodology : Literature Review," vol. 1, no. 2, hal. 85–114, 2022.
- D. Syaripudin dan A. Walad, "ALGORITMA RANDOM FOREST PADA DATA PREDIKSI KECACATAN PERANGKAT LUNAK RESAMPLING TECHNIQUE TO INCREASE THE ACCURACY VALUE OF RANDOM FOREST ALGORITHM ON SOFTWARE DEFECT PREDICTION," hal. 4845–4858, 2024.
- F. A. Setiawan, S. D. Putra, dan D. Sahlinal, "PENGUJIAN PROYEK WEBSITE OTOMATISASI DENGAN PENDEKATAN INTEGRASI ANTARA SELENIUM DAN TESTNG," hal. 1–14, 2015.
- F. Nawir dan A. W. A. Rivai, "EVALUASI TINGKAT USABILITAS WEB MOBILE PERGURUAN TINGGI NEGERI," vol. 1, no. 1, hal. 16–24, 2021.
- F. Yang, M. Minat, dan P. Mobile, "Faktor yang mempengaruhi minat penggunaan mobile banking," vol. 3, no. 17, 2021.
- H. Shafwanto, "Analisis User Experience Pada Website Informatika UNSIKA di Perangkat Mobile Menggunakan Metode Usability Testing," vol. 3, hal. 338–350, 2023.
- M. Sofyan, C. B. Sanjaya, D. T. Informatika, dan U. Y. Pasuruan, "Rancang bangun aplikasi pemesanan paket wisata di desa edelweiss wonokitri," vol. 12, no. 3, hal. 4080–4087, 2024.