

PEMBUATAN APLIKASI PELAPORAN MASALAH BERBASIS *MOBILE* MENGGUNAKAN *FRAMEWORK FLUTTER* (STUDI KASUS: PT BANK PEMBANGUNAN DAERAH MALUKU DAN MALUKU UTARA)

Sean Alessandro Pattirane, Pratyaksa Ocsa Nugraha Saian

Universitas Kristen Satya Wacana, Salatiga, Indonesia

Email: 672018112@student.uskw.edu, pratyaksa.ocsa@uksw.edu

Abstrak

PT Bank Pembangunan Daerah Maluku dan Maluku Utara (BPDM) merupakan bank pembangunan daerah yang berbentuk badan usaha milik pemerintah daerah yang berdiri pada tanggal 25 Oktober 1961. Dalam kegiatan operasional di PT BPDM, sering kali terjadi masalah pada saat kegiatan operasional berlangsung. Masalah-masalah yang terjadi adalah seperti kerusakan pada jaringan internal perusahaan, malfungsi pada aplikasi *core banking system*, dan juga masalah-masalah lainnya. Akan tetapi, pegawai di PT BPDM mengalami kesulitan dalam melaporkan dan menyimpan masalah-masalah tersebut. Berdasarkan masalah yang dijelaskan, maka penelitian ini dilakukan untuk mencari solusi terkait permasalahan tersebut, yaitu dengan membuat sebuah aplikasi pelaporan masalah dengan menggunakan *Framework Flutter*. Penelitian ini akan menggunakan metode penelitian *Waterfall*, dengan tahapan-tahapannya yaitu *Requirement and Analysis*, *System Design*, *Implementation*, *Testing* dan *Deployment*. Berdasarkan *user testing* menggunakan skala Likert menghasilkan persentase nilai rata-rata dari nilai indeks sebesar 82,27% yang artinya aplikasi pelaporan masalah ini mendapatkan tanggapan yang baik dari calon pengguna aplikasi.

Kata Kunci: *Flutter*, *Web Service*, Pelaporan Masalah

Abstract

PT Bank Pembangunan Daerah Maluku dan Maluku Utara (BPDM) is a regional development bank in the form of a regional government-owned business entity that was established on October 25, 1961. In operational activities at PT BPDM, problems often occur during operational activities. The problems that occur are such as damage to the company's internal network, malfunctions in the core banking system application, and also other problems. However, employees at PT BPDM have difficulty reporting and storing these problems. Based on the problems described, so this research was conducted to find solutions to these problems, namely by creating a problem reporting application using the Flutter Framework. This research will use the Waterfall research method, with the stages, namely Requirements and Analysis, System Design, Implementation, Testing and Deployment. Based on user testing using a Likert scale, the percentage of the average value of the index value is 82.27%, which means that this problem reporting application gets a good response from potential application users.

How to cite:	Sean Alessandro Pattirane, Pratyaksa Ocsa Nugraha Saian (2022). Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara). <i>Syntax Literate; Jurnal Ilmiah Indonesia</i> . 7 (8).
E-ISSN:	2548-1398
Published by:	Ridwan Institute

Keywords: *Flutter, Web Service, Problem Reporting*

Pendahuluan

PT Bank Pembangunan Daerah Maluku dan Maluku Utara (BPDM) merupakan bank pembangunan daerah yang berbentuk badan usaha milik pemerintah daerah. PT BPDM berdiri pada tanggal 25 Oktober 1961. Dalam kegiatan operasional PT BPDM, sering kali terjadi masalah saat kegiatan operasional dilakukan. Masalah-masalah tersebut dilaporkan untuk dicari solusi yang dapat digunakan untuk menyelesaikan permasalahan tersebut. Akan tetapi, PT BPDM kesulitan dalam menerima laporan masalah yang dilaporkan dan melakukan manajemen terhadap masalah yang dilaporkan. PT BPDM juga kesulitan untuk menyimpan masalah-masalah yang dilaporkan.

Berdasarkan masalah yang dijelaskan, dapat diketahui bahwa PT BPDM membutuhkan sebuah media yang dapat digunakan untuk melaporkan masalah dan menyimpan masalah-masalah tersebut. Maka dari itu, penelitian ini dilakukan untuk mencari solusi yang dapat menyelesaikan permasalahan tersebut, yaitu dengan membangun sebuah aplikasi pelaporan masalah.

Aplikasi pelaporan masalah harus dapat digunakan dengan mudah dan cepat, sehingga aplikasi dibuat berbasis *mobile*. Agar pembangunan aplikasi dapat dilakukan dengan lebih cepat dan efisien, maka dibutuhkan sebuah *framework* untuk membantu pembangunan aplikasi. *Framework* yang cocok untuk pembangunan aplikasi pelaporan masalah adalah *Flutter*.

Oleh karena itu, perlu dibuat sebuah aplikasi pelaporan masalah berbasis *mobile* dengan menggunakan *Framework Flutter*. *Flutter* merupakan *framework* pembuatan aplikasi *multi-platform* yang dibuat oleh Google menggunakan bahasa pemrograman *Dart* (Taufiq, Pratama, & Pratama, 2021); (Santoso, Suharso, & Hariyady, 2020). *Flutter* sebagai sebuah *framework* pembuatan aplikasi mempunyai kelebihan, yaitu bersifat *open source*, memiliki performa yang bagus dan proses *development* yang cepat (Andrian et al., n.d.). Penelitian ini akan menghasilkan sebuah aplikasi bernama aplikasi pelaporan masalah yang akan digunakan sebagai media pelaporan masalah pada kegiatan operasional PT BPDM.

Penelitian-penelitian yang mendukung penelitian ini antara lain adalah penelitian yang berjudul “Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten Pekalongan“, menjelaskan tentang sebuah sistem yang akan menjadi wadah bagi masyarakat Pekalongan untuk melaporkan masalah, kritikan atau keluhan terhadap pemerintah kabupaten Pekalongan. Pembangunan sistem pelaporan dibuat menggunakan bahasa pemrograman PHP dengan menggunakan *Framework CodeIgniter* (Permatasari, Priyambadha, & Arwan, 2018). Keterkaitan penelitian ini dengan penelitian yang akan dilakukan adalah sama-sama memiliki persamaan dalam kasus yang dibahas, yaitu terkait aplikasi pelaporan masalah. Sedangkan perbedaannya terletak pada *platform* dimana aplikasi akan berjalan, yaitu pada penelitian ini aplikasi akan berjalan pada *platform web*. Sedangkan, untuk penelitian yang akan dilakukan,

aplikasi akan berjalan pada *platform mobile*. Selain itu, perbedaan lainnya juga terdapat pada calon pengguna aplikasi, yaitu pada penelitian ini calon pengguna aplikasi adalah masyarakat di kabupaten Pekalongan. Sedangkan, untuk penelitian yang akan dilakukan, calon pengguna aplikasi adalah pegawai di PT BPDM.

Penelitian lain yang mendukung penelitian ini adalah penelitian berjudul “Perancangan Aplikasi Todolist Berbasis Android Menggunakan Flutter SDK dan Database Sqlite”, menjelaskan tentang sebuah aplikasi berbasis Android yang dapat digunakan oleh *user* untuk membuat daftar pekerjaan yang harus diselesaikan. Pembuatan aplikasi ini menggunakan *Framework Flutter* dan bahasa pemrograman *Dart* (Azmi & Rahmawati, 2021). Keterkaitan penelitian ini dengan penelitian yang akan dilakukan adalah sama-sama membahas tentang pembangunan sebuah aplikasi menggunakan *Framework Flutter*. Sedangkan perbedaannya terletak pada kasus yang diteliti, yaitu pada penelitian ini kasus yang dibahas adalah mengenai pembuatan Aplikasi Todolist. Sedangkan, untuk penelitian yang akan dilakukan, kasus yang dibahas adalah mengenai pembuatan aplikasi pelaporan masalah.

Berdasarkan penelitian-penelitian terdahulu yang membahas tentang pembuatan aplikasi untuk melaporkan masalah dan pembangunan sebuah aplikasi menggunakan *Framework Flutter*, maka akan dilaksanakan penelitian tentang pembuatan aplikasi pelaporan masalah berbasis *mobile* Menggunakan *Framework Flutter*.

Flutter merupakan *framework* pengembangan aplikasi yang menerapkan prinsip *single-code based* yang membuat *Flutter* masuk dalam jenis *framework* lintas *platform*. *Single-code base* memungkinkan aplikasi yang dibuat dengan *Flutter* dapat dijalankan pada *platform* yang berbeda dengan satu basis *code* saja (Aziz et al., 2020). *Flutter* dikembangkan menggunakan bahasa pemrograman *Dart*, sehingga bahasa pemrograman utama yang digunakan untuk membangun aplikasi pelaporan masalah adalah bahasa pemrograman *Dart*.

Dart adalah bahasa pemrograman yang dibuat oleh Google yang ditujukan untuk membangun aplikasi pada berbagai *platform*. *Dart* juga menggunakan konsep *Object Oriented Programming* (OOP) yang dimana pemrograman di dalamnya terdiri dari objek-objek tertentu (Wiryosukiro, 2019).

Aplikasi pelaporan masalah yang dibangun pada penelitian ini akan beroperasi pada *platform mobile*, yaitu *Android*. *Android* merupakan sistem operasi yang berkembang pesat pada beberapa tahun terakhir. *Android* bersifat gratis untuk digunakan yang dibuat untuk perangkat *mobile* (Nurhalimah, Suhartono, & Cahyana, 2017).

Database yang digunakan dalam penelitian ini untuk pengembangan aplikasi pelaporan masalah adalah *MySQL*. *MySQL* adalah jenis *database relational* yang menggunakan *Structured Query Language* (SQL) untuk pengelolaan *database*. *Database relational* berarti pengelolaan data-data dalam *database* dimasukkan ke dalam tabel-tabel supaya manipulasi data dapat dilakukan dengan lebih efisien (Maulana, 2017).

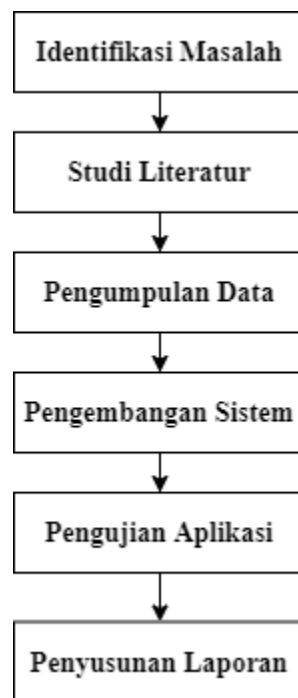
Penghubung *database* dan aplikasi pada aplikasi pelaporan masalah menggunakan teknologi *Web Service*. *Web Service* adalah teknologi yang

memungkinkan pembagian data antara perangkat lunak berbeda yang dengan menggunakan protokol HTTP. *Web Service* biasanya direpresentasikan dalam format JSON atau XML (Gunawan & Rahmatulloh, 2019); (Rizal & Rahmatulloh, 2019).

Web Service yang dibangun pada aplikasi pelaporan masalah menggunakan bahasa pemrograman *PHP Hypertext Protocol*. *PHP Hypertext Protocol* (PHP) merupakan bahasa pemrograman yang digunakan dalam pengembangan web. PHP bersifat gratis untuk digunakan dan hanya dijalankan prosesnya pada bagian server, tidak dijalankan pada *client* (Hidayat, Hartono, & Sukiman, 2017).

Metode Penelitian

Metode penelitian merupakan tahapan-tahapan dalam melakukan penelitian yang digambarkan secara sistematis yang bertujuan untuk memudahkan peneliti dalam melakukan penelitian. Bagan metodologi penelitian pembuatan aplikasi pelaporan masalah menjadi seperti pada **Gambar 1** (Edryanto, Irawan, & Effiyaldi, 2020).



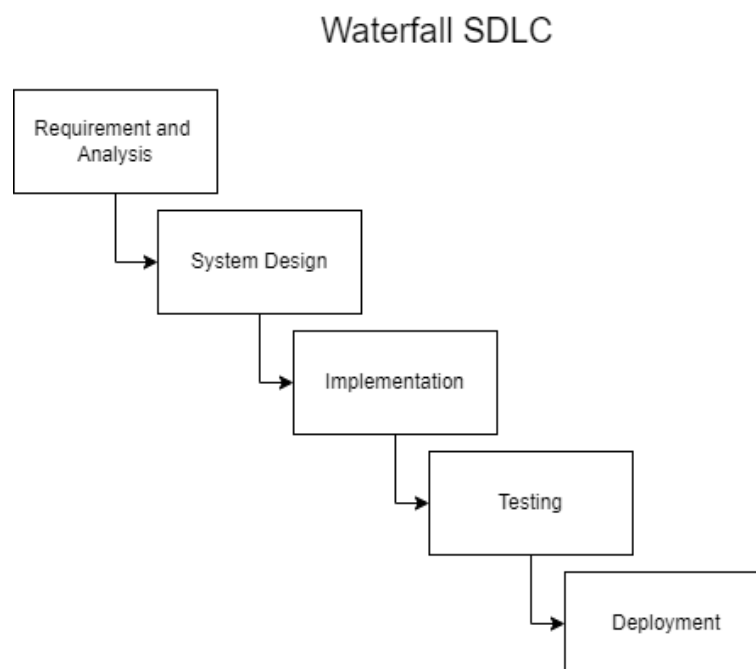
Gambar 1
Bagan Metode Penelitian

Tahapan-tahapan yang dilakukan pada penelitian ini ditampilkan pada **Gambar 1** yang dijelaskan sebagai berikut: Tahap Identifikasi Masalah merupakan tahapan yang digunakan untuk mempelajari masalah yang akan diteliti. Permasalahan yang ada adalah kesulitan PT BPDM dalam melaporkan masalah dan menyimpan data-data hasil pelaporan masalah yang terjadi di PT BPDM; Tahap Studi Literatur merupakan tahapan yang digunakan untuk melakukan pengumpulan literatur review yang digunakan dalam penelitian ini. Beberapa literatur review yang digunakan yaitu *Flutter*, *Dart* dan *MySQL*; Tahap Pengumpulan Data merupakan tahapan yang digunakan untuk mencari data yang diperlukan untuk penelitian ini, yaitu data daftar kantor cabang di PT BPDM;

Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)

Tahap Pengembangan Sistem merupakan tahapan dimana sistem akan dibangun, mulai dari pembuatan Unified Modelling Language (UML) yang merupakan gambaran dari desain sistem aplikasi yang akan dibuat. Selanjutnya, UML tersebut akan dijadikan patokan dalam pembuatan aplikasi. Pembuatan aplikasi dimulai dengan pembuatan fungsi-fungsi yang akan dihubungkan ke *user interface*. Setelah itu, ada pembuatan *database* dengan *MySQL* dan pembuatan *web service* dengan menggunakan bahasa pemrograman PHP. Selanjutnya, dilakukan proses untuk menggabungkan semua sistem menjadi satu kesatuan; Tahap Pengujian Aplikasi adalah tahap dimana dilakukan pengujian terhadap aplikasi yang telah dibuat, untuk mencari berapa banyak *error* dan *bug* yang ada di dalam aplikasi ini; Tahap Penyusunan Laporan merupakan tahapan terakhir dari penelitian yang dilakukan.

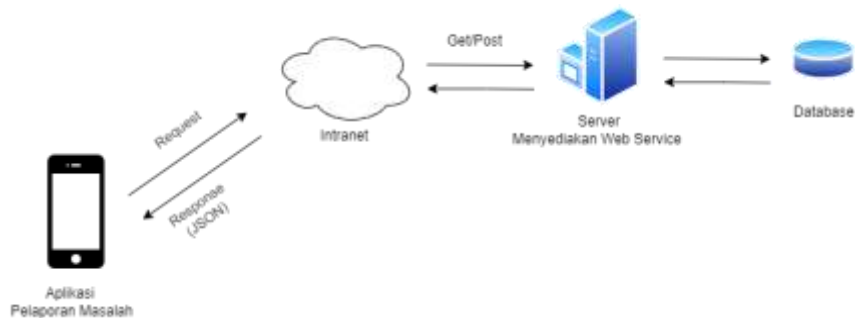
Aplikasi pelaporan masalah memiliki tahapan-tahapan yang teratur dalam proses pengembangannya. Maka dari itu, *Waterfall* menjadi *Software Development Life Cycle* (SDLC) yang dipilih dalam pengembangan aplikasi pelaporan masalah. *Waterfall* adalah SDLC yang dimana tahapan-tahapan yang dilakukan dalam pembuatan aplikasi bersifat *sequence* dan berkelanjutan. Berikut ini merupakan bagan *Waterfall* SDLC yang ditunjukkan pada **Gambar 2** (Sasmito, 2017).



Gambar 2
Bagan *Waterfall* SDLC

Tahapan yang dilakukan dalam *Waterfall* SDLC dalam pembangunan aplikasi pelaporan masalah adalah sebagai berikut: Tahap *Requirement and Analysis* yang merupakan tahapan yang dilakukan untuk mengumpulkan dan menganalisis kebutuhan aplikasi pelaporan masalah yang sesuai dengan keinginan *user*. Selanjutnya tahap *System Design* yang dilakukan untuk membuat arsitektur sistem dan desain sistem yang

akan digunakan. **Gambar 3** merupakan gambar arsitektur sistem dari aplikasi pelaporan masalah.



Gambar 3
Arsitektur Sistem

Gambar 3 merupakan gambar arsitektur sistem dari aplikasi pelaporan masalah yang akan dibuat. Aplikasi pelaporan masalah tidak dapat melakukan akses langsung ke *database*. Pertama-tama, aplikasi pelaporan masalah harus mengirimkan *request* terlebih dahulu ke *web service*. *Request* dapat dikirimkan dengan menggunakan metode *get* atau *post*. Setelah itu, *web service* akan mengambil data dari *database* berdasarkan *request* yang dikirimkan. Hasil data dari *database* akan diubah menjadi JSON dan kemudian akan dikirimkan ke aplikasi pelaporan masalah sebagai *response* dari *request* yang dilakukan oleh aplikasi. Sedangkan, untuk desain sistem aplikasi digambarkan menggunakan *Usecase*, *Class* dan *Activity Diagram* yang ditampilkan pada **Gambar 4**, **Gambar 5** dan **Gambar 6**.

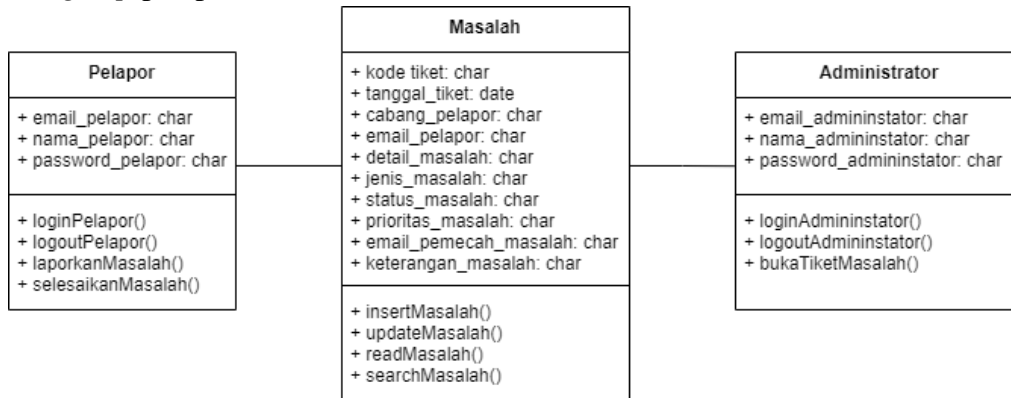


Gambar 4
Usecase Diagram

Pada **Gambar 4** terdapat dua aktor pada aplikasi pelaporan masalah, yaitu pelapor dan administrator. Pelapor maupun administrator harus melakukan *login*

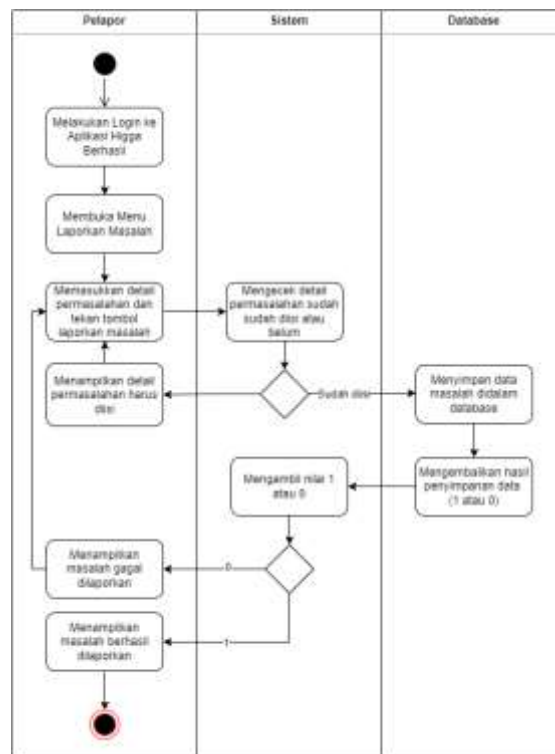
Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)

terlebih dahulu sebelum bisa menggunakan semua fitur yang disediakan pada aplikasi pelaporan masalah. Satu-satunya fitur yang dapat diakses sebelum melakukan *login* adalah *sign up* pelapor.



Gambar 5
Class Diagram

Pada **Gambar 5** terdapat tiga *class* yang terdapat pada *Class Diagram*, yaitu Masalah, Pelapor dan Administrator. Setiap *class* terbagi atas tiga bagian, yaitu komponen atas yang berisi nama dari *class*, komponen tengah yang berisi atribut dari *class* dan komponen bawah yang berisi metode dari *class*. Dapat dilihat bahwa masing-masing *class* memiliki atribut dan metodenya masing-masing. Selain itu, terdapat hubungan asosiasi antara *class* Administrator dan Pelapor dengan *class* Masalah.



Gambar 6
Activity Diagram

Gambar 6 merupakan *Activity Diagram* yang menggambarkan proses pelaporan masalah yang dilakukan oleh pelapor. Pertama-tama, pengguna aplikasi harus melakukan *login* sebagai pelapor supaya bisa membuka menu laporkan masalah. Setelah menu laporkan masalah terbuka, pelapor harus memasukkan detail permasalahan dan menekan tombol laporkan masalah. Jika detail permasalahan belum diisi, maka akan menampilkan detail permasalahan harus diisi. Jika detail permasalahan sudah diisi, maka sistem akan menyimpan data ke dalam *database* dan *dabatase* akan mengembalikan nilai 1 atau 0 yang diterima oleh sistem. Jika nilai kembalian adalah 0, maka akan menampilkan masalah gagal dilaporkan. Jika nilai kembalian adalah 1, maka akan menampilkan masalah berhasil dilaporkan.

Setelah tahap *System Design*, selanjutnya dilakukan tahap *Implementation* yang dimana desain *database*, desain sistem dan arsitektur sistem yang telah dibuat, diubah menjadi sebuah aplikasi dengan melakukan proses pengkodean. Proses pengkodean aplikasi pelaporan masalah menggunakan *Framework Flutter*. Selanjutnya, tahap *Testing* akan dilakukan untuk menguji aplikasi pelaporan masalah yang telah dibuat. Pengujian dilakukan dengan menggunakan metode *Black Box Testing* dan *User Testing*. Setelah tahap *Testing* telah selesai dilakukan, maka tahapan terakhir akan dilakukan yaitu tahap *Deployment*. Pada tahap *Deployment*, aplikasi pelaporan masalah akan mulai digunakan di PT BPDM.

Hasil dan Pembahasan

Penelitian yang dilakukan oleh peneliti menghasilkan sebuah aplikasi pelaporan masalah yang dibuat dengan menggunakan *Framework Flutter* yang berjalan pada *platform mobile*, yaitu *Android*. Berikut ini merupakan tampilan aplikasi yang telah dibuat:



Gambar 7
Splash Screen

Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)

Gambar 7 adalah halaman *splash screen* aplikasi yang merupakan tampilan yang pertama kali muncul saat pengguna membuka aplikasi pelaporan masalah.



Gambar 8
Sign Up

Gambar 8 adalah halaman *sign up* aplikasi yang merupakan halaman untuk mendaftarkan akun dengan jenis pengguna pelapor, jika belum memiliki akun.



Gambar 9
Login

Gambar 9 adalah halaman *login* aplikasi yang merupakan halaman untuk melakukan *login*. Halaman ini dapat digunakan oleh pelapor maupun administrator dengan memasukkan *credential login* dan memilih jenis pengguna. Jika *login* berhasil,

jenis pengguna pelapor akan diarahkan ke beranda pelapor, sedangkan jenis pengguna administrator akan diarahkan ke beranda administrator.



Gambar 10
Beranda Pelapor

Gambar 10 adalah halaman beranda pelapor. Halaman beranda pelapor berisi data statistik dari masalah yang telah dilaporkan dan penanganan masalah yang harus diselesaikan oleh pelapor.



Gambar 11
Halaman Laporkan Masalah

Gambar 11 adalah halaman laporkan masalah yang digunakan oleh pelapor untuk melaporkan masalah-masalah yang terjadi saat kegiatan operasional dilaksanakan.

Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)



Gambar 12
Halaman Daftar Masalah

Gambar 12 adalah halaman daftar masalah. Halaman ini adalah halaman yang terdapat pada jenis pengguna pelapor dan administrator. Pada administrator, halaman ini menampilkan semua masalah yang telah dilaporkan oleh pelapor. Sedangkan pada pelapor, halaman ini menampilkan masalah yang dilaporkan oleh pelapor tersebut saja.



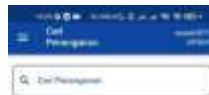
Gambar 13
Halaman Cari Masalah

Gambar 13 adalah halaman cari masalah. Halaman cari masalah juga terdapat di jenis pengguna pelapor dan administrator. Halaman ini digunakan untuk mencari data masalah berdasarkan *keyword* yang dimasukkan. Halaman ini akan menampilkan data masalah sesuai dengan jenis pengguna.



Gambar 14
Halaman Daftar Penanganan

Gambar 14 adalah halaman daftar penanganan yang terdapat pada jenis pengguna pelapor. Halaman ini berisi daftar masalah yang harus ditangani oleh pelapor tertentu.



Gambar 15
Halaman Cari Penanganan

Gambar 15 adalah halaman cari penanganan pada jenis pengguna pelapor yang dapat digunakan untuk mencari data penanganan masalah yang harus ditangani oleh pelapor tertentu.

Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)



Gambar 16
Beranda Administrator

Gambar 16 adalah beranda administrator yang dapat diakses oleh administrator untuk melihat data statistik masalah yang telah dilaporkan oleh pelapor. Data statistik yang dapat dilihat adalah seperti statistik berdasarkan status, jenis maupun prioritas masalah.

The screenshot shows the 'Buka Tiket Masalah' (Open Issue Ticket) page. It includes a header with the title 'Buka Tiket Masalah' and a user profile 'admin'. Below the header, there is a section for 'PROMOSI1432020' with details:

- Waktu: 24-05-2022 (K01)
- Gabung: Cabang Pasa
- Email Pelapor: p4w@p4w.com
- Jenis Masalah: Pelaporan (0A)
- Status Masalah: Status Dikaji
- Detail Masalah: Detail

Below the details, there is a section titled 'Memproses Tiket Masalah' (Process Issue Ticket) with two dropdown menus:

- Pemroses Masalah:** Pilih Pemroses Masalah
- Prioritas Masalah:** Pilih Prioritas Masalah

A 'Buka Tiket' button is located at the bottom of the form.

Gambar 17
Halaman Buka Tiket Masalah

Gambar 17 adalah halaman buka tiket masalah pada jenis pengguna admin. Halaman ini digunakan untuk memproses masalah yang telah dilaporkan oleh pelapor tertentu dan masalah tersebut akan diberikan kepada pelapor lainnya untuk ditangani.



Gambar 18
Halaman Detail Penanganan

Gambar 18 adalah halaman detail penanganan yang dapat digunakan oleh pelapor untuk memasukkan solusi terhadap masalah tertentu yang menjadi tanggung jawab untuk ditangani.



Gambar 19
Halaman Detail Masalah

Gambar 19 adalah halaman detail masalah. Halaman ini dapat diakses oleh jenis pengguna pelapor dan administrator untuk melihat detail masalah dari suatu masalah yang telah dilaporkan.

Dalam pembuatan aplikasi pelaporan masalah, penghubung antara aplikasi dan *database MySQL* menggunakan *web service*. *Web service* dibuat dengan menggunakan

Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)

bahasa pemrograman *PHP*. *Dio* menjadi *library* yang digunakan untuk melakukan *request* data dari aplikasi ke *web service*.

```
<?php
require("../config.php");
$response = array();
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $kode_tiket = $_POST["kode_tiket"];
    $tanggal_tiket = $_POST["tanggal_tiket"];
    $cabang_pelapor = $_POST["cabang_pelapor"];
    $email_pelapor = $_POST["email_pelapor"];
    $detail_masalah = $_POST["detail_masalah"];
    $jenis_masalah = $_POST["jenis_masalah"];
    $status_masalah = $_POST["status_masalah"];
    $query = "INSERT INTO masalah
        (kode_tiket, tanggal_tiket, cabang_pelapor,
        email_pelapor, detail_masalah,
        jenis_masalah, status_masalah)
        VALUES
        ('$kode_tiket', '$tanggal_tiket', '$cabang_pelapor',
        '$email_pelapor', '$detail_masalah',
        '$jenis_masalah', '$status_masalah')";
    $exe = mysqli_query($connection, $query);
    $check = mysqli_affected_rows($connection);
    if($check > 0){
        $response["kode"] = 1;
        $response["pesan"] = "Berhasil Melaporkan Masalah";
    }else{
        $response["kode"] = 0;
        $response["pesan"] = "Gagal Melaporkan Masalah";
    }
}
else{
    $response["kode"] = 0;
    $response["pesan"] = "Tidak Ada Post Data";
}
echo json_encode($response);
mysqli_close($connection);
?>
```

Kode Program 1

Kode Program untuk Laporkan Masalah pada *Web Service*

Kode Program 1 adalah kode program pada *web service* untuk melakukan pelaporan masalah. Kode program ini akan menerima *request* dari aplikasi dengan metode *post*. Kemudian data dari *request* tersebut akan dimasukkan kedalam *database*. Selanjutnya, kode program akan memberikan *response* dalam bentuk JSON, sesuai dengan berhasil atau gagalnya data dimasukkan kedalam *database*. Kode program ini juga akan mengembalikan *response*, jika metode *request* yang digunakan oleh aplikasi bukan menggunakan metode *post*.


```

static Future<List> reportProblemFunction(
    kodeTiket,
    tanggalTiket,
    cabangPelapor,
    emailPelapor,
    detailMasalah,
    jenisMasalah,
    statusMasalah,
) async {
    FormData formData = FormData.fromMap({
        "kode_tiket": kodeTiket,
        "tanggal_tiket": tanggalTiket,
        "cabang_pelapor": cabangPelapor,
        "email_pelapor": emailPelapor,
        "detail_masalah": detailMasalah,
        "jenis_masalah": jenisMasalah,
        "status_masalah": statusMasalah,
    });
    var response = await dio.post(reportProblemPelaporURL, data: formData);

    int jsonKode = json.decode(response.data) ["kode"];
    String jsonPesan = json.decode(response.data) ["pesan"];

    List<dynamic> data = [];

    data.add(jsonKode);
    data.add(jsonPesan);

    return data;
}

```

Kode Program 2 **Kode Program untuk Fungsi Laporkan Masalah pada Aplikasi**

Kode Program 2 adalah kode program untuk fungsi laporkan masalah pada aplikasi. Kode program ini berupa sebuah fungsi dengan tujuh parameter. Saat melakukan *request* kepada *web service* untuk melakukan *login* pelapor, ketujuh parameter tersebut akan dikirimkan. Kemudian hasil *response* dalam bentuk JSON dari *web service*, akan didekode menjadi dua variabel dan ditambahkan ke dalam sebuah *list*. *List* tersebut kemudian akan menjadi kembalian dari fungsi ini.

Untuk mengetahui apakah aplikasi pelaporan masalah telah berfungsi dengan baik, maka diperlukan sebuah pengujian terlebih dahulu sebelum nanti aplikasi pelaporan masalah ini digunakan oleh PT BPDM. Tujuan dari pengujian ini adalah untuk mencari apakah ada *bug* yang dapat membuat aplikasi mengalami kerusakan. Metode pengujian yang dilakukan adalah dengan menggunakan *Black Box Testing*. *Black Box Testing* merupakan metode pengujian yang hanya berfokus terhadap fungsional dari suatu aplikasi tanpa harus memeriksa struktur kode dari aplikasi tersebut (Aini, N., Wicaksono, S. A., & Arwani, 2019). Hasil pengujian dengan menggunakan metode *Black Box Testing* disajikan dalam **Tabel 1**, **Tabel 2** dan **Tabel 3**.

Tabel 1
Testing Fungsi Login Pelapor

Kasus dan Hasil Pengujian Benar			
Skenario Pengujian	Hasil yang ingin dicapai	Pengamatan	Kesimpulan

Pembuatan Aplikasi Pelaporan Masalah Berbasis Mobile Menggunakan Framework Flutter (Studi Kasus: PT Bank Pembangunan Daerah Maluku dan Maluku Utara)

Memasukkan dan memilih data <i>login</i> pelapor dengan benar	Data <i>login</i> pelapor cocok dengan data di <i>database</i> dan halaman berpindah ke beranda pelapor	Halaman berpindah ke beranda pelapor	(✓) Diterima () Ditolak
Kasus dan Hasil Pengujian Salah			
Skenario Pengujian	Hasil yang ingin dicapai	Pengamatan	Kesimpulan
Memasukkan dan memilih data <i>login</i> pelapor dengan salah	Menampilkan pesan <i>error username</i> atau <i>password</i> salah	Muncul pesan <i>error username</i> atau <i>password</i> salah	(✓) Diterima () Ditolak

Tabel 1 merupakan tabel pengujian *Black Box Testing* untuk fungsi *login* pelapor. **Tabel 1** membuktikan bahwa fungsi *login* pelapor telah sesuai dengan kebutuhan pengguna karena saat pengguna memasukkan dan memilih data *login* pelapor dengan benar, maka halaman aplikasi akan dialihkan ke menu beranda pelapor. Sedangkan, jika data *login* pelapor dimasukkan dengan salah, maka akan muncul pesan *error* bahwa *username* atau *password* salah.

Tabel 2
Testing Menu Laporkan Masalah

Kasus dan Hasil Pengujian Benar			
Skenario Pengujian	Hasil yang ingin dicapai	Pengamatan	Kesimpulan
Memasukkan detail permasalahan yang terjadi dan tekan tombol laporkan masalah	Data masalah ditambahkan ke <i>database</i> dan menampilkan pesan berhasil laporkan masalah	Muncul pesan berhasil laporkan masalah	(✓) Diterima () Ditolak
Kasus dan Hasil Pengujian Salah			
Skenario Pengujian	Hasil yang ingin dicapai	Pengamatan	Kesimpulan
Tidak memasukkan detail masalah dan tekan tombol laporkan masalah	Menampilkan pesan <i>error</i> detail masalah harus diisi	Muncul pesan <i>error</i> detail masalah harus diisi	(✓) Diterima () Ditolak

Tabel 2 merupakan tabel pengujian *Black Box Testing* untuk menu laporkan masalah. **Tabel 2** membuktikan bahwa menu laporkan masalah telah sesuai dengan kebutuhan pengguna karena saat pengguna memasukkan detail permasalahan dan tekan tombol laporkan masalah, maka akan muncul pesan berhasil melaporkan masalah. Sedangkan, jika pengguna tidak memasukkan detail permasalahan, maka akan muncul pesan *error* bahwa detail permasalahan harus diisi.

Tabel 3
Testing Menu Buka Tiket Masalah

Kasus dan Hasil Pengujian Benar			
Skenario Pengujian	Hasil yang ingin dicapai	Pengamatan	Kesimpulan

Memilih pemecah masalah dan prioritas masalah, lalu tekan tombol buka tiket	Data masalah akan di <i>update</i> pada <i>database</i> dan menampilkan pesan tiket berhasil dibuka	Muncul pesan tiket berhasil dibuka	(✓) Diterima () Ditolak
Kasus dan Hasil Pengujian Salah			
Skenario Pengujian	Hasil yang ingin dicapai	Pengamatan	Kesimpulan
Tidak memilih pemecah masalah dan prioritas masalah, lalu tekan tombol buka tiket	Menampilkan pesan <i>error</i> tiket gagal dibuka	Muncul pesan <i>error</i> tiket gagal dibuka	(✓) Diterima () Ditolak

Tabel 3 merupakan tabel pengujian *Black Box Testing* untuk menu buka tiket masalah. **Tabel 3** membuktikan bahwa menu buka tiket masalah telah sesuai dengan kebutuhan pengguna. Karena saat pengguna memilih pemecah masalah dan prioritas masalah lalu tekan tombol laporkan masalah, maka akan muncul pesan tiket berhasil dibuka. Sedangkan, jika pengguna tidak memilih pemecah masalah dan prioritas masalah, maka akan muncul pesan *error* bahwa tiket gagal dibuka.

Hasil pengujian aplikasi menggunakan metode *Black Box Testing* pada aplikasi pelaporan masalah menunjukkan bahwa tidak ditemukan adanya *error* atau *bug* yang dapat mengganggu kinerja aplikasi pelaporan masalah. Akan tetapi, sebelum aplikasi dapat digunakan oleh pengguna, diperlukan sebuah pengujian dari sisi pengguna (*user testing*). Pengujian *user testing* ini dilakukan untuk mengetahui tanggapan pengguna terkait desain, kemudahan dalam mengoperasikan aplikasi dan kegunaan aplikasi untuk mengatasi permasalahan terkait pelaporan masalah di PT BPDM. Pengujian *user testing* ini melibatkan 25 calon pengguna aplikasi. Berikut adalah pertanyaan dan jawaban dari calon pengguna aplikasi lewat kuesioner yang telah diberikan:

Tabel 4
Pertanyaan dan Jawaban Kuesioner

No	Pertanyaan	Jawaban				
		STS	TS	C	S	SS
1.	Aplikasi pelaporan masalah yang telah dibuat dapat dengan mudah dipahami oleh pengguna?	-	-	5	7	13
2.	Aplikasi pelaporan masalah yang telah dibuat telah sesuai dengan kebutuhan pengguna?	-	-	6	13	6
3.	Aplikasi yang telah dibuat dapat mempermudah dan mempercepat pelaporan masalah yang terjadi?	-	-	7	4	14
4.	Aplikasi pelaporan masalah mempunyai tampilan yang baik dan menarik?	-	-	11	10	4
5.	Aplikasi pelaporan masalah dapat menyimpan masalah-masalah yang terjadi?	-	-	2	12	11

Tabel 4 adalah hasil dari kuesioner yang telah disebar pada calon pengguna. Setelah itu, peneliti melakukan pemberian skor atau bobot kepada setiap pilihan jawaban telah diberikan lewat kuesioner. Pemberian skor jawaban dapat dilihat pada **Tabel 5** (Pranatawijaya, Widiatry, Priskila, & Putra, 2019).

Tabel 5
Skor Jawaban

Jawaban	Hasil Skor
Sangat Setuju	5
Setuju	4
Cukup	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Setelah pemberian skor telah dilakukan, proses selanjutnya adalah menghitung skor maksimum yang akan diperlukan untuk menghitung nilai indeks. Skor maksimum dapat dihitung dengan melakukan perkalian jumlah responden terhadap skor tertinggi Likert, yaitu $25 \times 5 = 125$. Setelah itu, perhitungan data dilakukan yang dapat dilihat pada **Tabel 6**.

Tabel 6
Hasil Perhitungan Skala Likert

Pertanyaan	Nilai Skor					Total Skor	Nilai Indeks (Total Skor/ Skor Maksimum) x 100) %
	STS (1 Skor)	TS (2 Skor)	C (3 Skor)	S (4 Skor)	SS (5 Skor)		
1.	-	-	5	7	13	108	86,4%
2.	-	-	6	13	6	100	80%
3.	-	-	7	4	14	107	85,6%
4.	-	-	11	10	4	93	74,4%
5.	-	-	2	12	11	109	87,2%

Tabel 7
Range Penilaian

Indikator Range	Hasil
0% - 19,99%	Sangat Tidak Setuju
20% - 39,99%	Tidak Setuju
40% - 59,99%	Kurang Setuju
60% - 79,99%	Setuju
80 - 100%	Sangat Setuju

Tabel 6 menunjukkan bahwa nilai rata-rata dari nilai indeks adalah sebesar 82,72% yang dapat dikategorikan “Sangat Setuju” jika berpatokan pada *range* penilaian

yang terdapat pada **Tabel 7**. Jika melihat persentase dari rata-rata nilai indeks, dapat diambil kesimpulan bahwa aplikasi pelaporan masalah mendapatkan tanggapan yang baik dari calon pengguna aplikasi.

Kesimpulan

Penelitian ini berhasil menghasilkan sebuah solusi untuk PT BPDM dalam mengatasi permasalahan yang terkait dengan kesulitannya pegawai di PT BPDM dalam melaporkan dan menyimpan masalah-masalah yang terjadi. Solusi yang dihasilkan berupa sebuah aplikasi yang bernama aplikasi pelaporan masalah, yang dibangun dengan menggunakan *Framework Flutter* yang berjalan pada *platform mobile*, yaitu *Android*. Berdasarkan hasil pengujian *user testing* menggunakan skala Likert yang mendapatkan persentase nilai rata-rata dari nilai indeks sebesar 82,27%, dapat diambil kesimpulan bahwa aplikasi pelaporan masalah di PT BPDM mendapatkan tanggapan yang baik dari calon pengguna aplikasi.

BIBLIOGRAFI

- Aini, N., Wicaksono, S. A., & Arwani, I. (2019). Pembangunan Sistem Informasi Perpustakaan Berbasis Web menggunakan Metode Rapid Application Development (RAD)(Studi pada: SMK Negeri 11 Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(9), 8647–8655. [Google Scholar](#)
- Andrean, Kenji, Informatika, Departemen, Sains, Institut, Surabaya, Terpadu, Armanto, Hendrawan, Informatika, Departemen, Sains, Institut, Surabaya, Terpadu, Pickerling, C., Informatika, Departemen, Sains, Institut, & Surabaya, Terpadu. (n.d.). *Sistem Tempat Parkir Terintegrasi yang Dilengkapi dengan Aplikasi Mobile dan Mikrokontroller*. 22–29.
- Aziz, Dendy Armandiaz, Andreswari, Rachmadita, Fajar, Soni, Gumilang, Surya, Industri, Fakultas Rekayasa, & Telkom, Universitas. (2020). Perancangan Bisnis Dan Arsitektur Aplikasi Pada Aplikasi Mobile Manawa Investasi Hewan Ternak Business and Application Architecture Design in Manawa. *E-Proceeding of Engineering*, 7(2), 7111–7121. [Google Scholar](#)
- Azmi, Rifky, & Rahmawati. (2021). Perancangan Aplikasi Todolist Berbasis Android Menggunakan Flutter Sdk Dan Database Sqlite. *Jurnal Seminar Nasional Riset Dan Teknologi*, 567–572. [Google Scholar](#)
- Edryanto, Irawan, & Effiyaldi. (2020). Perancangan Aplikasi Augmented Reality Pengenalan Landmark Asia Tenggara Berbasis Android Pada Sdn 139/Iv Jambi. *Jurnal Ilmiah Mahasiswa Teknik Informatika*, 2(1), 31–42. [Google Scholar](#)
- Gunawan, Rohmat, & Rahmatulloh, Alam. (2019). JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 5(1), 74. <https://doi.org/10.26418/jp.v5i1.27232> [Google Scholar](#)
- Hidayat, Harri, Hartono, & Sukiman. (2017). Pengembangan Learning Management System (LMS) Untuk Bahasa Pemrograman PHP. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 5(1), 20–29. [Google Scholar](#)
- Maulana, Muhammad Sahidin Rizal. (2017). Rancang Bangun Aplikasi Kasir Tiket Nonton Bola Bareng Pada X Kasir Di Suatu Lokasi X Dengan Visual Basic 2010 Dan MYSQL Ninuk. *Ekp*, 13(3), 1576–1580. [Google Scholar](#)
- Nurhalimah, Septina Restu, Suhartono, Suhartono, & Cahyana, Ucu. (2017). Pengembangan Media Pembelajaran Mobile Learning Berbasis Android pada Materi Sifat Koligatif Larutan. *JRPK: Jurnal Riset Pendidikan Kimia*, 7(2), 160–167. <https://doi.org/10.21009/jrpk.072.10> [Google Scholar](#)
- Permatasari, Richa Amalia, Priyambadha, Bayu, & Arwan, Achmad. (2018). Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten

Pekalongan. *J-Ptiik*, 2(11), 5604–5610. [Google Scholar](#)

Pranatawijaya, Viktor Handrianus, Widiatry, Widiatry, Priskila, Ressa, & Putra, Putu Bagus Adidyana Anugrah. (2019). Penerapan Skala Likert dan Skala Dikotomi Pada Kuesioner Online. *Jurnal Sains Dan Informatika*, 5(2), 128–137. <https://doi.org/10.34128/jsi.v5i2.185> [Google Scholar](#)

Rizal, Randi, & Rahmatulloh, Alam. (2019). *RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan*. (Xml). [Google Scholar](#)

Santoso, Hartanto, Suharso, Wildan, & Hariyady, Hariyady. (2020). Pembangunan Aplikasi Mobile Hybrid Pada M-Voting Pemilu Raya Universitas Muhammadiyah Malang. *Indonesian Journal of Applied Informatics*, Vol. 4, p. 127. <https://doi.org/10.20961/ijai.v4i2.40870> [Google Scholar](#)

Sasmito, Ginanjar Wiro. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 2(1), 6–12. [Google Scholar](#)

Taufiq, Arif, Pratama, M., & Pratama, Ahmad R. (2021). Rancang Bangun Aplikasi Android “Kuliah Apa?” Berbasis Flutter dan TensorFlow Lite. *Automata*, 2(1). [Google Scholar](#)

Wiryosukiro, Wagiman. (2019). Belajar Bahasa Pemrograman Dart Untuk Pemula – Part 1. *Emka.Web.Id*, (1), 2–5.

Copyright holder:

Sean Alessandro Pattirane, Pratyaksa Ocsa Nugraha Saian (2022)

First publication right:

Syntax Literate: Jurnal Ilmiah Indonesia

This article is licensed under:

