

Rancang Bangun Aplikasi *Android* pada *Section Engineering* di *Head Office*

Anis Choirunnisa¹⁾

Teknologi Informasi, Politeknik Gajah Tunggal
anis@poltek-gt.ac.id

M. Ridwan Arif Cahyono²⁾

Teknik Elektronika, Politeknik Gajah Tunggal
ridwan@poltek-gt.ac.id

Margareta Shinta Herawati³⁾

Teknik Elektronika, Politeknik Gajah Tunggal
margaretashinta2564@gmail.com

ABSTRAK

Proses pengajuan Formulir Perintah Kerja (FPK) secara manual di *Section Engineering* menyebabkan berbagai kendala seperti keterlambatan pengiriman, kesulitan pencatatan, serta rendahnya efisiensi dan akurasi data. Penelitian ini bertujuan untuk merancang dan membangun sistem digital berupa *Aplikasi Android* berbasis kodular untuk mempermudah *tenant* untuk membuat *request work order* ke pihak *Engineering*. Antarmuka pengguna dirancang menggunakan *tools* draw.io dan sistem *backend* menggunakan *PHP* dan *MYSQL*. Aplikasi ini memungkinkan pengguna untuk mengakses informasi dan mengajukan permintaan kerja secara praktis melalui perangkat *mobile*. Hasil uji menunjukkan efisiensi waktu yang meningkat sebesar 70,36% dan efisiensi biaya mencapai 82,43% dibandingkan dengan sistem manual. Selain itu, fitur-fitur seperti *Form Complain*, *Dashboard*, *Survey Satisfaction*, Instruksi dan Profil membantu mempercepat alur penggunaan dan memperkecil resiko kehilangan data. Sistem ini juga telah diuji menggunakan metode *black box* dan menunjukkan hasil yang baik. Dengan demikian, sistem digital ini efektif mendukung transformasi digital dalam proses kerja engineering.

Kata Kunci: Formulir Perintah Kerja, Kodular, Aplikasi *Android*, *PHP*, *Engineering*.

ABSTRACT

The manual process of submitting Work Order Forms (FPK) in the Engineering Section causes various obstacles such as late delivery, difficulty in recording, and low efficiency and data accuracy. This study aims to design and build a digital system in the form of a Kodular-based Android Application to make it easier for tenants to make work order requests to Engineering. The user interface is designed using the draw.io tool and the backend system uses PHP and MYSQL. This application allows users to access information and submit work requests practically through mobile devices. Test results show an increase in time efficiency of 70.36% and cost efficiency reaching 82.43% compared to the manual system. In addition, features such as the Complaint Form, Dashboard, Satisfaction Survey, Instructions and Profiles help speed up the user flow and minimize the risk of data loss. This system has also been tested using the black box method and shows good results. Thus, this digital system effectively supports digital transformation in engineering work processes.

Key Words: Work Order Form, Kodular, Android Application, PHP, Engineering.

I. PENDAHULUAN

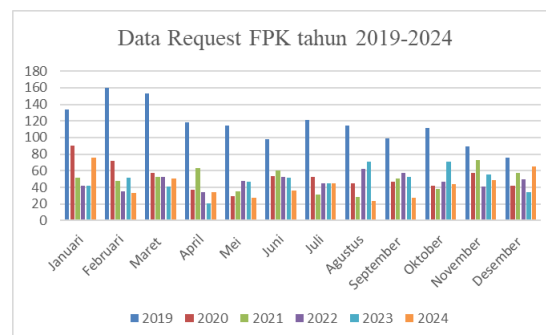
Semakin berkembangnya zaman, inovasi teknologi termasuk dalam bidang manajemen properti dan fasilitas, terus mengalami kemajuan yang pesat. Hubungan antara manajemen gedung dan penyewa menjadi kunci utama dalam operasionalisasi dan kelangsungan kerja sama dalam bisnis penyewaan gedung. Hal ini melibatkan komunikasi yang efektif, penanganan keluhan secara tepat waktu, dan pemeliharaan lingkungan yang nyaman[1]. Namun hingga kini, masih banyak proses pelaporan dan penanganan keluhan (service request) yang mencakup perbaikan fasilitas, kebersihan, maupun keluhan operasional lainnya yang berjalan secara manual, salah satunya di *Building Management departement* divisi *Engineering*. Salah satu sistem yang masih berjalan secara manual yakni Formulir Perintah Kerja (FPK) yang terdapat pada proses kerja perawatan dan perbaikan alat serta gedung. Ketergantungan pada sistem manual tersebut mengakibatkan sejumlah kendala seperti keterlambatan, kehilangan informasi, serta kurangnya transparansi dalam proses penanganan keluhan. Kekurangan lain dari sistem manual antara lain yakni kurangnya efisien karena sistem manual seringkali memakan banyak waktu dibandingkan sistem yang telah berjalan secara otomatis. Rekapitulasi data secara manual rentan terhadap kesalahan (*human error*) yang berdampak pada tingkat akurasi informasi, juga mempersulit pekerja dalam proses pengolahan data.



Gambar 1. Kondisi Penyimpanan Berkas

Seperti yang terlihat pada Gambar 1, penyimpanan data fisik juga memiliki banyak kekurangan lain yakni diantaranya membutuhkan ruang penyimpanan yang relatif besar, resiko berkas hilang bahkan berkas rusak. Banyaknya data yang disimpan secara manual juga mempersulit pekerja dalam mengakses berkas, bahkan terkadang sulit ditemukan pada saat data sedang dibutuhkan.

Terdapat juga data dari penelitian yang telah kami lakukan, sebagai berikut.



Gambar 2. Data Kebutuhan Kertas FPK per Tahun

Data diatas menunjukkan banyaknya penggunaan kertas dalam 6 tahun (2019-2024) yang berjumlah total sebanyak 16.393 FPK, dimana pada 1 rangkap FPK memiliki 3 lembar kertas yakni kertas putih yang ditujukan untuk arsip data *Engineering*, kertas biru ditujukan untuk TRO/Sec/BM, dan kertas berwarna pink yang ditujukan untuk *Tenant* atau pelapor. Data diatas menghasilkan rata-rata sebanyak 2.732 FPK per tahun.

Tabel 1. Harga Kertas per FPK

Komponen	Satuan	QTY	Biaya	
			Satuan	Total
Harga Buku			Rp	
FPK (150 lembar)	Buku	19	25,000	Rp 475,000
Kertas per				Rp
FPK (3 rangkap)	Request	2732	Rp 500	1,366,000
Total Biaya Manual per Tahun				Rp 1,841,000

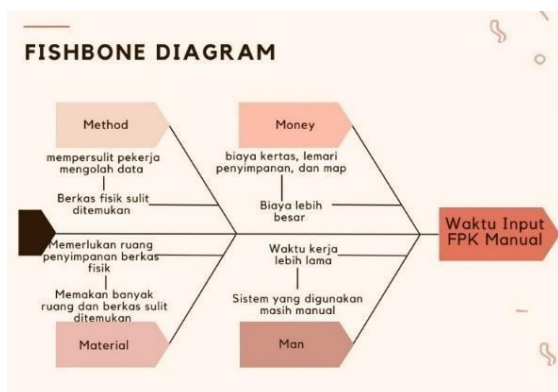
Pada satu buku FPK terdapat 50 rangkap FPK berjumlah 150 lembar kertas. Harga 1 buku FPK bernilai 25.000 rupiah.

Tabel 2. Data Waktu Input FPK Manual

No	Waktu Proses	No	Waktu Proses
1	504	16	503
2	525	17	543
3	470	18	504
4	518	19	482
5	472	20	500
6	554	21	490
7	480	22	468
8	484	23	475

9	535	24	512
10	511	25	517
11	524	26	519
12	474	27	512
13	495	28	474
14	471	29	503
15	465	30	550
Rata-rata	501.13	Rata-rata	501.13

Dari 30 sampel yang telah didapat, menghasilkan rata rata waktu dalam satu kali penginputan FPK secara manual, yakni selama 501,13 detik. Sehingga kami juga mencoba untuk menganalisa permasalahan di atas menggunakan metode *Fishbone Diagram* :



Gambar 3. *Fishbone Diagram*

Berdasarkan analisa yang didapat dari *Fishbone Diagram*, penelitian ini merujuk pada permasalahan yakni tidak adanya sistem digital untuk formulir perintah kerja sehingga seluruh proses kerja masih dilakukan secara manual. Salah satu solusi yang digunakan adalah sistem *digital* untuk mempermudah proses kerja menjadi lebih efisien. Sistem ini dapat mengurangi waktu kerja menjadi lebih singkat dan mengubah sistem manual menjadi otomatis sehingga mempermudah para pekerja untuk berkomunikasi dan membuat proses kerja menjadi lebih efisien, dari segi waktu ataupun tenaga.

Sistem yang kami bangun adalah *Website* berbasis Codeigniter dan Aplikasi Android berbasis Kodular. Kami membuat Aplikasi Android dilakukan dengan mempertimbangkan beberapa faktor utama, salah satunya adalah tingkat adopsi pengguna yang tinggi di lingkungan operasional perusahaan. Mayoritas pengguna di lapangan menggunakan perangkat Android, sehingga pengembangan aplikasi pada platform ini menjadi pilihan yang lebih efisien dan tepat sasaran.

Terdapat juga alternatif bagi pengguna dengan perangkat yang tidak kompatibel yakni dengan cara membuat request *Form Complain* secara manual melalui pihak engineering yang selanjutnya akan dilakukan pengInputan *Form Complain* melalui *Website* oleh admin engineering.

Berdasarkan hal tersebut, penelitian yang kami jalankan bertujuan untuk merancang dan membangun sistem digital untuk Formulir Perintah Kerja yang dapat berfungsi secara otomatis agar dapat mempermudah para pekerja dalam menjalankan proses kerja, meningkatkan efisiensi serta mengurangi resiko kesalahan yang mungkin muncul.

Perancangan sistem digital menggunakan metode UML dapat membantu dalam menggambarkan dan merancang struktur serta fungsi sistem secara menyeluruh. UML adalah bahasa yang digunakan untuk menspesifikasikan, memvisualisasikan, dan membuat struktur dasar sistem perangkat lunak. Ini juga memungkinkan pemodelan aturan bisnis [2]. UML terdiri dari beberapa diagram yang digunakan untuk merancang atau mengembangkan perangkat lunak berbasis objek, antara lain *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

Use Case Diagram merupakan visualisasi dari suatu sistem yang terdiri dari beberapa komponen, seperti *Actor*, use case, dan hubungan antar komponen [3]. *Activity Diagram* merupakan diagram yang digunakan untuk memvisualisasikan proses paralel yang terjadi ketika sistem dijalankan [4]. *Sequence Diagram* membantu dalam memahami persyaratan sistem baru, mendokumentasikan proses, dan memvisualisasikan skenario teknis saat sistem berjalan, memungkinkan pengguna memprediksi perilaku sistem [5]. Sedangkan *Class Diagram* menggambarkan struktur dan hubungan antar *class*, *package*, dan *objek*, termasuk pewarisan dan asosiasi.

Terdapat beberapa penelitian terdahulu, diantaranya M. Ahmadar membuat sistem aplikasi untuk mendukung berkembangnya teknologi informasi saat ini agar proses penjualan dan pemasaran bisa tersebar luas [6]. Adapun S. Eko membuat perancangan Aplikasi Pengenalan Budaya Nusantara Di Indonesia Berbasis Android yang dibuat untuk memudahkan bagi para penggunanya

dalam mengingat pengetahuan tentang budaya nusantara yang ada di Indonesia [2]. E. Trivaika and M. A. Senubekti yang membuat sistem aplikasi mobile yang akan menampilkan data keuangan baik pemasukan maupun pengeluaran untuk menghindari perilaku konsumtif masyarakat [7]. Sementara itu, Rismayanti, dkk membuat produk berupa e-modul berbasis kodular pada smartphone yang valid, praktis serta efektif dalam meningkatkan kemampuan literasi matematis siswa [8]. U. Kholifah dan N. Imansari membuat pemrograman Kodular agar aktifitas mereka dalam penggunaan smart phone dapat diarahkan pada hal yang positif [9]. N. K. Hendriawan, dkk membuat pengembangan media pembelajaran Kodular untuk belajar merupakan sebuah media bantu pembelajaran yang sesuai dengan kebutuhan pembelajaran di sekolah, khususnya di SMK Pasundan 1 Kota Serang. 1 [10]. S. Salsabila, dkk sadar bahwa kurangnya variasi media pembelajaran berakibat rendahnya kemampuan matematis peserta didik dan penelitian ini bertujuan untuk menghasilkan produk berupa e-modul pada android menggunakan kodular yang layak, serta efektif dalam meningkatkan kemampuan representasi matematis siswa [11]. R. Hermiati, dkk membuat Aplikasi E-commerce ini dapat membantu pelanggan untuk mendapatkan informasi tentang produk yang ditawarkan, proses membeli dan menjual serta mempermudah pelanggan dalam proses pemesanan suatu produk, dan dapat membantu meningkatkan penjualan bagi toko [12]. Terdapat juga sistem informasi penjualan yang dikembangkan oleh M. Ahmadar menggunakan bahasa pemrograman PHP dengan menggunakan MySQL sebagai database [6].

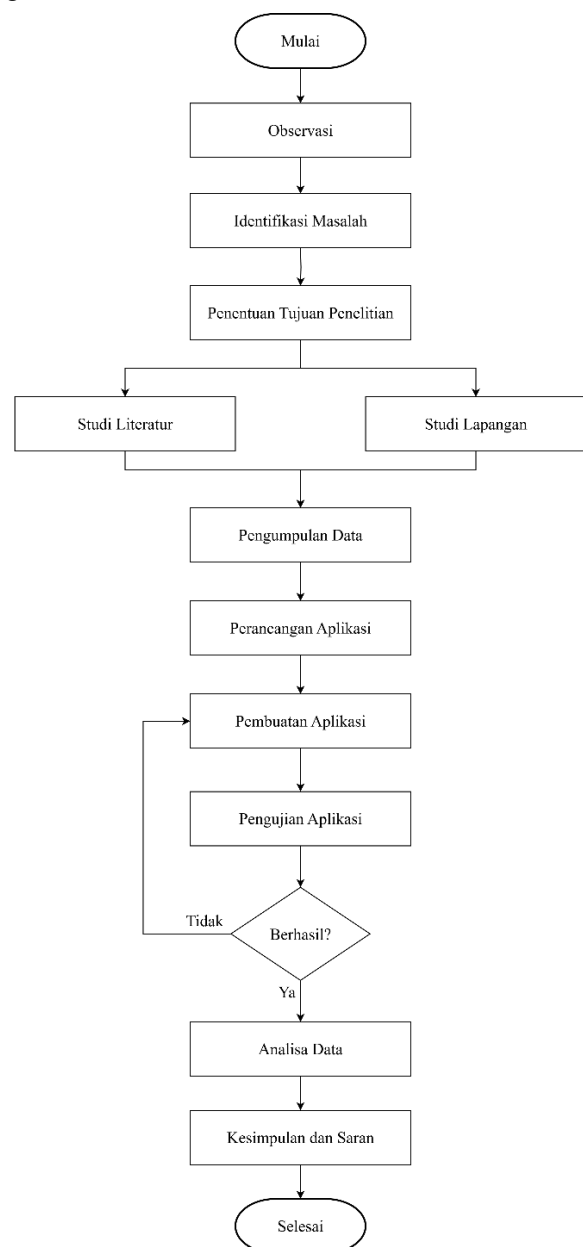
Menurut A. A. Nozari, saat ini terdapat beberapa sistem di PT A berupa Aplikasi Mobile Android di Server Hosting [13]. A. C. Praniffa, melakukan pengujian sistem yang dilakukan menggunakan blackbox testing dan whitebox testing yang menghasilkan sistem informasi parkir pada universitas islam negeri sultan syarif kasim riau tidak ditemukan permasalahan dari sistem yang dibangun [14]. Pengujian aplikasi/software yang dilakukan oleh E. Novalia and A. Voutama merupakan tahapan untuk memeriksa apakah suatu program berjalan dengan semestinya atau belum salah satunya pengujian dilakukan menggunakan Black Box [15]. Pengujian perangkat lunak merupakan salah satu tahapan yang paling penting dalam pengembangan sebuah sistem informasi maka

perlu dilakukan pengujian dengan menggunakan black box testing [16]. Dalam uji coba yang telah dilakukan mencapai hasil yang sesuai tidak terjadi downtime, apabila terjadi kegagalan pada salah satu server database maka dibuatkan hosting untuk database server [17].

II. METODE PENELITIAN

A. Alur Penelitian

Alur penelitian ini dilakukan untuk merancang Aplikasi berbasis *Android* Formulir Perintah Kerja (FPK) yang dapat dilihat pada gambar berikut :



Gambar 4. Flowchart Alur Penelitian

1) Observasi

Observasi merupakan langkah awal dalam melakukan penelitian dengan cara mengamati proses secara langsung kondisi lapangan untuk mencari informasi terkait fenomena yang ingin diteliti dan dianalisa. pada tahapan ini, observasi dilakukan pada *Section Engineering* dengan mengamati proses penerimaan *Complain* dari *Tenant* / pelapor, pengisian permintaan pada *form* perintah kerja yang dilanjutkan dengan penanganan *Complain* pelapor yang dilakukan oleh teknisi, proses pengisian dan penyerahan FPK oleh *Engineer* kepada Admin.

2) Identifikasi Masalah

Hal ini dilakukan dengan mengamati bagaimana proses perintah kerja berlangsung, mulai dari penerimaan keluhan dari *Tenant*, pengisian FPK pada bagian permintaan oleh Admin, penanganan keluhan oleh *Engineer*, pengisian FPK oleh *Engineer*, dan kembalinya FPK ke Admin sebagai arsip data. Setelah mengidentifikasi masalah ditemukan bahwa masih terdapat banyak sistem yang berjalan secara manual dan menggunakan kertas, salah satunya Formulir Perintah Kerja (FPK). Hal tersebut menyebabkan penumpukan *form* pada ruang penyimpanan. Penumpukan ini juga menimbulkan resiko hilangnya berkas atau bahkan rusak serta sulitnya proses pencarian data.

3) Penentuan Tujuan Penelitian

Berdasarkan observasi dan identifikasi masalah yang dilakukan oleh penulis, penelitian ini bertujuan untuk mempermudah proses perintah kerja dengan merancang bangun digitalisasi Formulir Perintah Kerja (FPK) berbasis aplikasi.

4) Studi Literatur dan Studi Lapangan

Studi literatur adalah metode pengumpulan data dan informasi serta sumber pendukungnya, seperti buku, catatan, hasil penelitian sebelumnya, dan artikel-artikel ilmiah [18]. Studi literatur dilakukan oleh penulis untuk memperkuat kajian serta memperkaya wawasan sehingga memudahkan penulis dalam melakukan penelitian. Sedangkan studi lapangan merupakan metode untuk mendapatkan data dan informasi dengan melakukan observasi dan wawancara secara langsung [19]. Studi lapangan bertujuan untuk mengetahui sistem dan cara kerja formulir perintah kerja, serta memahami kondisi yang sebenarnya terjadi.

5) Pengumpulan Data

Pengumpulan data merupakan teknik atau cara yang digunakan oleh penulis untuk memperoleh informasi yang dibutuhkan dalam penelitian [20]. Pada tahap ini, penulis melakukan pengumpulan data yang diperlukan dalam penelitian. Adapun data yang dikumpulkan mencakup data waktu penginputan FPK ke dalam *excel* oleh Admin, data jumlah FPK setiap bulan dalam kurun waktu 5 tahun ke belakang, data karyawan pada *Section Engineering*, data daftar *Tenant* di *Head Office*, dan data *cost* yang dikeluarkan untuk penggunaan kertas FPK.

6) Perancangan Desain Aplikasi

Penulis membuat perancangan desain aplikasi untuk menentukan *layout* tombol, *interface*, dan fitur yang akan dibuat. Perancangan dilakukan dengan menggunakan metode *Unified Modelling Language* (UML). UML digunakan untuk memvisualisasikan suatu sistem agar mempermudah dalam proses perancangan.

7) Pembuatan Aplikasi

Pada tahap ini dilakukan pembuatan aplikasi *Request Job* untuk *Tenant*. Pembuatan aplikasi didasarkan pada desain yang sudah dirancang menggunakan metode UML. Pembuatan aplikasi pada penelitian ini menggunakan *Kodular* sebagai media untuk membuat aplikasi dan *MySQL* sebagai basis data.

8) Pengujian Aplikasi

Pada tahap ini dilakukan pengujian aplikasi *Request Job* untuk *Tenant*. Pengujian aplikasi menggunakan *black box testing*. Metode *black box testing* dipilih karena berfokus pada fitur fungsi dan memastikan fitur aplikasi berjalan dengan baik dan alur proses aplikasi sesuai dengan proses kerja FPK.

9) Analisa Data

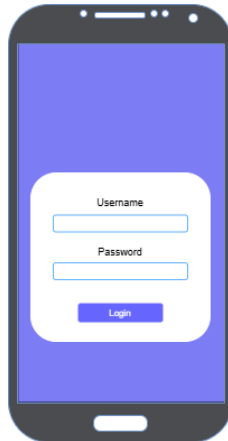
Analisa data merupakan proses yang terdiri dari berbagai tindakan, seperti mengurai, membedakan, memilah, dan mengolah data untuk mengklasifikasikan kembali data berdasarkan kriteria tertentu yang kemudian akan dihasilkan sebuah kesimpulan. Pengolahan data-data ini dilakukan dengan merekap lamanya waktu yang dibutuhkan selama proses kerja. Analisis ini juga dilakukan menguji apakah terdapat perbedaan waktu terhadap lamanya proses kerja sebelum dan sesudah aplikasi tersebut diterapkan.

10) Kesimpulan dan Saran

Dari data yang sudah dianalisis sebelumnya maka akan diperoleh kesimpulan dan saran. Kesimpulan mencakup hasil analisis dan penelitian

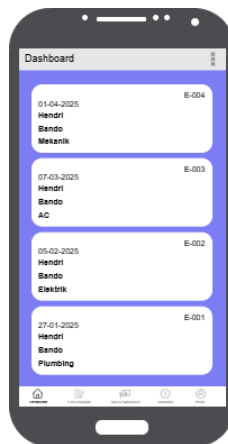
yang menjawab tujuan dari penelitian ini yang berguna untuk menyimpulkan hasil dari aplikasi yang sudah dibuat. Sedangkan saran mencakup hal-hal apa saja yang masih dapat dikembangkan dari penelitian yang telah dilakukan.

B. Perancangan Aplikasi



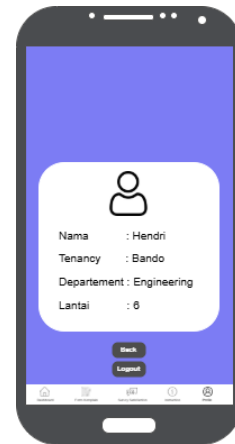
Gambar 5. Login Aplikasi

Berdasarkan gambar diatas, terdapat halaman *Login* untuk *User* sebelum masuk ke dalam aplikasi. Pada halaman *Login* terdapat *Username* akun *User* beserta *Password*. Terdapat pula tombol *Login* yang dapat *User* tekan ketika *Username* dan *Password* telah selesai diisi. *User* tidak dapat masuk ke aplikasi apabila *Username* dan *Password* yang dimasukkan salah.



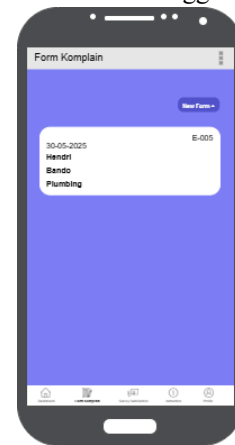
Gambar 6. Dashboard Aplikasi

Berdasarkan Gambar 6 terdapat halaman *Dashboard* yang menyajikan ringkasan informasi *history* FPK yang sudah selesai. Halaman *Dashboard* akan ditampilkan, jika *User* berhasil *Login*.



Gambar 7. Profile User Aplikasi

Gambar diatas merupakan desain halaman *Profile User* pada aplikasi. Halaman ini dirancang untuk menampilkan informasi pribadi *User*. Informasi pribadi pada halaman tersebut terdiri dari nama, *tenancy*, *departement*, dan lantai. Pada halaman ini pula, *User* dapat melakukan *Logout* dari aplikasi jika sudah selesai menggunakannya.



Gambar 8. Form Complain Aplikasi

Gambar 8 merupakan desain halaman *Form Complain* pada aplikasi. Halaman ini dirancang untuk menampilkan daftar keluhan yang diinput oleh *User* dalam bentuk *Cardview*. Selain menampilkan, *User* juga dapat membuat keluhan baru melalui tombol *New Form*. *Cardview Form Complain* terdiri dari No. FPK yang akan terisi otomatis, tanggal FPK, nama *Tenant*, nama pelapor dan jenis keluhan.

Gambar 9. *New Form Complain* Aplikasi

Gambar diatas menunjukkan desain halaman *New Form Complain* pada aplikasi. Halaman ini memungkinkan pengguna untuk menyampaikan keluhan atau laporan gangguan secara detail. Beberapa data yang ditampilkan secara otomatis dan tidak bisa diedit pada Form ini antara lain no. FPK, tanggal, nama pelapor, *departement*, lantai, dan nama *Tenant*. Pengguna dapat melengkapi beberapa data yang tersedia seperti lokasi, jenis keluhan, deskripsi keluhan, dan *capture* foto. Terdapat tombol *submit* untuk mengirim *Form Complain* dan tombol *cancel* untuk membatalkan keluhan.

Gambar 10. Detail *Form Complain* Aplikasi

Gambar 10 merupakan desain halaman detail *Form Complain* pada aplikasi yang berfungsi untuk menampilkan informasi lengkap mengenai laporan keluhan yang diajukan. Informasi yang ditampilkan pada halaman ini antara lain no.FPK, tanggal, status, nama pelapor, nama *Tenant*, *departement*, lokasi, lantai, jenis keluhan, deskripsi keluhan, dan *capture* foto.

Gambar 11. *Survey Satisfaction* Aplikasi

Gambar diatas menunjukkan desain halaman utama dari fitur *Survey Satisfaction* pada aplikasi *mobile*. Halaman ini memiliki fungsi utama untuk menampilkan daftar komplain yang sudah ditangani oleh *Engineer* dan siap untuk diberikan penilaian oleh pengguna. Daftar komplain pada halaman ini dirancang dalam bentuk *Cardview* yang menampilkan informasi seperti no.FPK, tanggal, nama pelapor, nama *Tenant*, dan jenis keluhan.

Gambar 12. Detail *Survey Satisfaction* Aplikasi

Gambar diatas merupakan desain Halaman *Survey Satisfaction*, dimana form ini wajib diisi oleh *User* apabila *Engineer* sudah selesai mengerjakan keluhan yang diajukan. Halaman ini merupakan penilaian dari hasil kerja yang telah diselesaikan *Engineer*. Terdapat pula catatan yang dapat diisi oleh *User* apabila perbaikan yang dilakukan belum sempurna/kurang baik.

III. HASIL DAN PEMBAHASAN

A. Tampilan Antarmuka Aplikasi

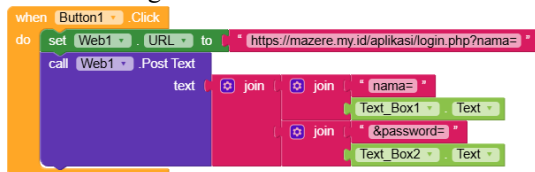
a) Login



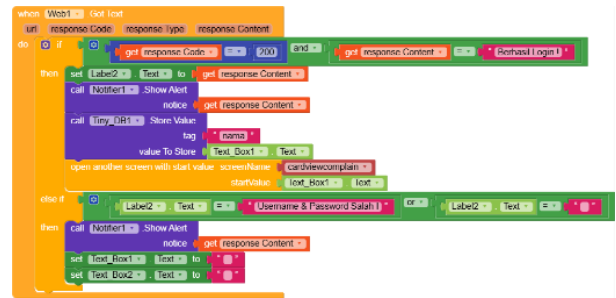
Gambar 13. Tampilan Halaman Login

Gambar 13 merupakan tampilan pada halaman *Login*. Pada halaman ini terdapat komponen *text box1* yang berfungsi sebagai tempat untuk mengisi nama dan terdapat *text box2* untuk tempat mengisi *password* oleh pengguna. Pada halaman ini, pengguna memerlukan *username* dan *password* sesuai dengan akun yang sudah didaftarkan oleh pihak *Engineering*. Terdapat juga tombol *Login* yang apabila tombol tersebut ditekan maka program PHP akan mencari akun di *database* yang sesuai dengan akun yang dimasukkan pengguna.

Apabila *username* dan *password* yang dimasukkan pengguna sesuai dengan akun yang terdaftar di *Database*, maka aplikasi akan berpindah ke halaman *Home Form Complain*. Namun apabila *username* dan *password* yang dimasukkan tidak sesuai dengan akun yang terdaftar maka akan muncul notifikasi peringatan bahwa *username* dan *password* salah. Berikut merupakan blok program untuk fitur login:



Gambar 14. Blok Program Submit Login



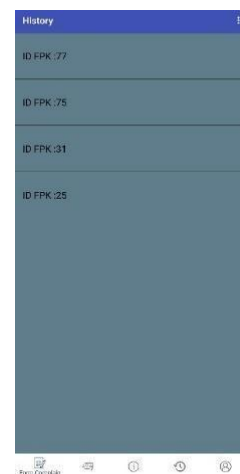
Gambar 15. Blok Program Read Database

Berikut merupakan program php untuk fitur login:

```
$login = mysqli_query($conn, "SELECT * FROM users WHERE nama='$nama' AND password='$password'");
$cek = mysqli_num_rows($login);
if ($cek > 0) {
    $data = mysqli_fetch_assoc($login);
    if ($data['permission'] == 4) {
        echo "Berhasil Login !";
    } else {
        echo "Silahkan login dengan akun tenant";
    }
} else {
    echo "Username & Password Salah !";
}
```

Program diatas merupakan logika untuk membaca data di database kolom users, apabila permission akun = 4 maka berhasil login, apabila permission akun selain 4, maka tidak bisa login.

b) Dashboard History



Gambar 16. Dashboard History

Halaman *Dashboard* merupakan tampilan yang dirancang untuk menampilkan riwayat aktivitas atau data yang telah terjadi sebelumnya dalam suatu sistem aplikasi. Pada halaman *Dashboard* terdapat komponen *list view* yang berfungsi untuk menampilkan data no FPK yang sudah selesai

dikerjakan ataupun yang sedang dikerjakan. Seluruh data no FPK akan tampil sesuai dengan nama akun yang sedang *Login*. Data yang ditampilkan yakni no FPK yang apabila ditekan akan menuju ke halaman detail *History Dashboard* dan menampilkan detail lebih lanjut mengenai data *Work Order*. Berikut merupakan blok program kodular dari fitur dashboard :



Gambar 17. Blok Program Get Data History

Berikut merupakan program php dari fitur history:
\$query = "SELECT * FROM work_order
WHERE nama = '\$nama' ORDER BY LENGTH
(id_fpk) DESC, id_fpk DESC";

Program diatas berfungsi untuk mengambil seluruh data id_fpk yang ada berdasarkan nama akun yang sedang login. Apabila id_fpk di klik maka akan muncul data berdasarkan id_fpk yang dituju.

c) Detail *History Dashboard*



Gambar 18. Tampilan Halaman Detail History

Halaman detail *Dashboard* merupakan Halaman detail *Dashboard* merupakan halaman yang menampilkan informasi lengkap terkait *Work Order* sesuai dengan no FPK yang dipilih di list view pada halaman *History*. Pada halaman ini terdapat komponen label dan *image* yang berfungsi untuk menampilkan data detail *History*. Data yang ditampilkan pada halaman ini diantaranya no. FPK, nama pelapor, *departement*, lantai, lokasi, *tenancy*, no. FPK, status, jenis keluhan, deskripsi keluhan, foto keluhan, foto pengerjaan, foto selesai, tindakan penanggulangan, jam mulai, jam selesai pengerjaan, nama *Engineer*, *rating* dan *note*.

Data detail yang ditampilkan merupakan data yang ada di *database* menggunakan id_FPK sebagai acuan dalam pencarian data yang terhubung dengan data id_FPK yang ada di halaman *Dashboard*. Sehingga apabila list view di halaman *Dashboard* yang menampilkan id_FPK ditekan maka halaman aplikasi akan berganti menjadi halaman detail *History* dan menampilkan data sesuai dengan id FPK yang dituju. Berikut merupakan blok program kodular untuk mengambil data :



Gambar 19. Blok Program Detail History

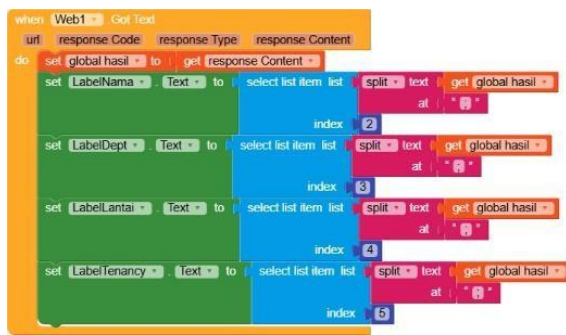
Berikut program php fitur history:
\$id_fpk = isset(\$_GET['id_fpk']) ?
mysqli_real_escape_string(\$conn,
\$_GET['id_fpk']) : '';
Program diatas berfungsi untuk menampilkan data di database sesuai dengan id_fpk yang dituju.

d) Profile



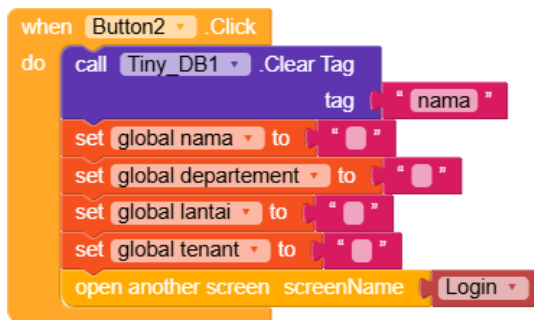
Gambar 20. Profile Aplikasi

Gambar 20 merupakan gambar halaman *Profile*. Halaman ini menampilkan data terkait akun pengguna yang sedang *Login* ke dalam aplikasi. Dalam halaman profil terdapat label nama yang menampilkan nama pengguna, label *departement*, terdapat juga label lantai yang menampilkan informasi mengenai keterangan lantai tempat pengguna bekerja. Dan terdapat label *tenancy* yakni informasi terkait nama perusahaan tempat pengguna bekerja. Berikut merupakan blok program kodular untuk tampil data profil :



Gambar 21. Blok Program Get Data Profil

Pada halaman ini, aplikasi menampilkan data yang sesuai dari variabel *username* & *password* yang terdaftar di *Database*, apabila terdapat kecocokan nama dan *password* maka program php akan menampilkan data *departement*, lantai dan *tenancy* dengan fitur *get*.



Gambar 22. Blok Program Logout

Terdapat juga tombol *Logout* dan *back*. Apabila pengguna menekan tombol *back* maka halaman aplikasi akan kembali ke halaman *Dashboard*. Sedangkan tombol *Logout* berfungsi untuk reset akun, dimana akun yang sedang *Login* akan keluar dan pengguna diminta untuk melakukan *Login* akun kembali.

Berikut php untuk tampil data profil:

```
$query_user = "SELECT * FROM users WHERE  
nama = '$user'";
```

```
echo $id_fpk . " , " . // nomor FPK baru dari tabel  
work_order
```

```
$data_user['nama'] . " , "  
$data_user['departement'] . " , "  
$data_user['lantai'] . " , "  
.$data_user['tenancy'];
```

```
} else {  
echo "User tidak ditemukan";}
```

<https://jurnal.poltek-gt.ac.id/index.php/jiti/>
Program Studi D3 Teknologi Informasi Politeknik Gajah Tunggal

Program diatas berfungsi untuk menampilkan data sesuai dengan akun yang sedang login.

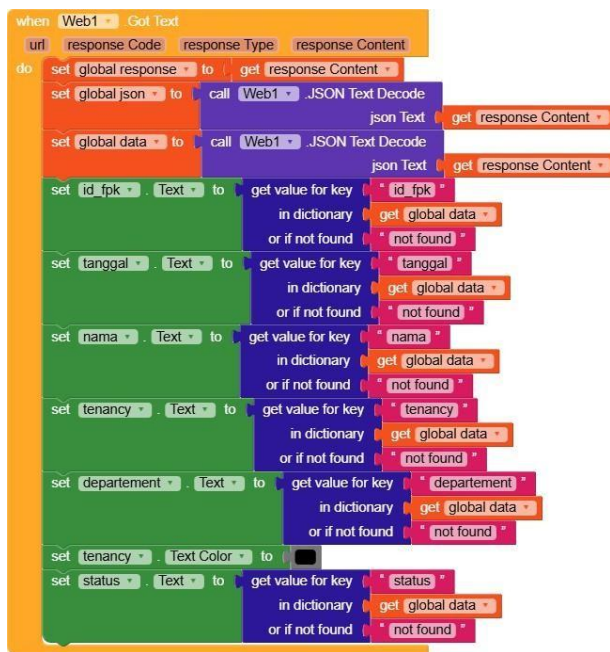
e) Home Form Complain



Gambar 23. Home Form Complain

Halaman *Home Form Complain* merupakan informasi utama yang dirancang untuk memudahkan pengguna dalam memantau aktivitas terkait laporan keluhan yang terakhir dibuat. Halaman ini menampilkan data *Work Order* terakhir yang dikerjakan/ di *request*.

Data yang ditampilkan yakni no FPK, tanggal *Form Complain* dibuat, status *Work Order* yang dimana pengguna dapat melihat status penanganan dari pengerjaan *request*, nama pelapor kerusakan, *departement* asal pelapor, *tenancy* atau nama perusahaan yang melapor. Data yang ditampilkan merupakan data *Work Order* dengan no FPK tertinggi sehingga akan menampilkan *Work Order* terbaru yang ada di *database* sesuai nama akun. Terdapat juga tombol "New Form" untuk membuka halaman *Form Complain* untuk membuat *request* baru. Berikut blok program web1 yang berfungsi untuk menampilkan data form complain sesuai dengan *id_fpk* terakhir di *database* dan sesuai akun yang sedang login.



Gambar 24. Blok Program Dashboard Complain

Berikut merupakan php untuk menampilkan data form complain yang terakhir di input sesuai dengan akun yang sedang login:

```
$nama = isset($_GET['nama']) ?  
mysqli_real_escape_string($conn,  
$_GET['nama']): '';  
  
if (empty($nama)) {  
  
http_response_code(400);  
  
echo json_encode(["error" => "Parameter  
nama tidak ditemukan"]); exit;}  

```

f) Form Complain

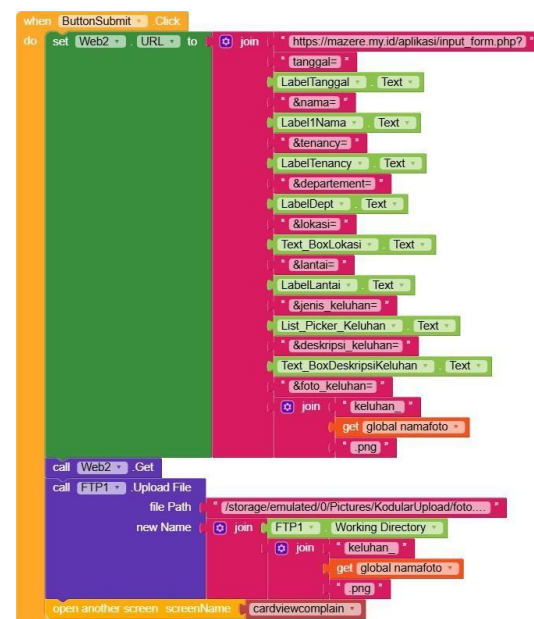
Gambar 25. New Form Complain

New Form Complain merupakan halaman untuk membuat *request Work Order* baru apabila terdapat kerusakan yang terjadi. Pada halaman *Form Complain* terdapat data yang terisi secara otomatis yakni nomor selanjutnya sesuai dengan no FPK terakhir di *database*. no FPK akan otomatis terisi

nomor selanjutnya sesuai nomor terakhir di *database* karena kolom *id_FPK* menggunakan *auto increment* sehingga tidak terdapat *request* dengan no FPK yang sama. Data tanggal juga terisi otomatis sesuai dengan *real time date* saat pengguna mengisi *Form Complain*. Selanjutnya terdapat data nama pelapor, *departement*, lantai dan *tenancy* yang akan terisi otomatis sesuai dengan akun yang sedang *Login*.

Pengguna hanya cukup mengisi lokasi kerusakan atau area kerusakan yang lebih spesifik, jenis keluhan (elektrik, mekanik, AC dan *Plumbing*), deskripsi keluhan, dan foto keluhan yakni foto kerusakan yang dapat dipilih dari galeri pengguna. Setelah semua data yang diminta sudah terisi, maka pengguna dapat menekan tombol *submit* untuk mengirim *Form Complain* pada pihak *Engineering* yang nantinya data tersebut akan masuk ke *database*. Pada saat dikirimkan, data status pada *request* akan berubah di *database* dari 0 menjadi 1 yang artinya *request* sudah diterima oleh pihak *Engineering* dan siap di proses.

Terdapat juga tombol *cancel* yang berfungsi untuk membatalkan *Form Complain* yang sudah diisi dan belum di *submit* sehingga apabila tombol ditekan maka halaman aplikasi akan membuka halaman *Home Form Complain* dan memperbarui halaman *Form Complain*. Berikut merupakan blok program untuk input data ke dalam *database* :



Gambar 26. Blok Program Input Data

Program php untuk fitur input data ke dalam database:

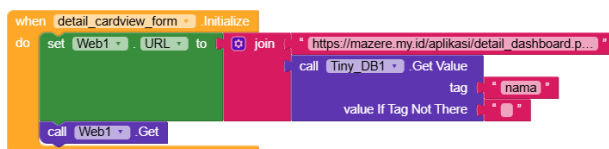
```
$sql = "INSERT INTO work_order (tanggal,  
status, id_tenant, nama, tenancy,  
departement, lokasi, lantai, jenis_keluhan,  
deskripsi_keluhan, tindakan_penanggulan  
gan, foto_keluhan, foto_pengerjaan,  
foto_selesai, jam_mulai, jam_selesai,  
id_engineer, engineer1, engineer2,  
engineer3, engineer4, admin_approved,  
sh_approved, rating, note)  
g) Detail Form Complain
```



Gambar 27. *Detail Form Complain*

Halaman *Detail Form Complain* merupakan tampilan lanjutan yang menyajikan informasi lengkap dari setiap laporan keluhan yang telah diajukan oleh pengguna. Pada halaman ini, seluruh data penting ditampilkan secara menyeluruh, mulai dari no FPK, status, tanggal, nama, *departement*, lokasi, lantai, jenis keluhan, deskripsi keluhan, foto keluhan, tindakan penanggulangan, foto pengerjaan, jam mulai, jam selesai, nama *Engineer*, foto selesai, *rating* dan *note*. Data yang ditampilkan di *detail Form Complain* merupakan data dengan no FPK terbesar yang ada di *Database* sesuai dengan nama akun yang sedang *Login*.

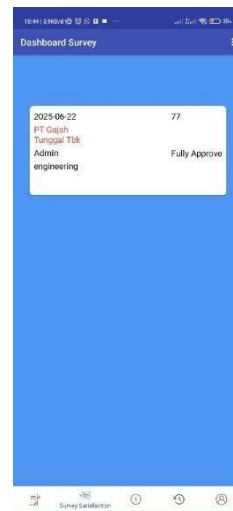
Berikut merupakan blok program untuk mengambil data form complain yang terakhir diinput.



Gambar 28. Blok Program Get Data Detail Complain

Berikut program php yang berfungsi untuk mengambil data form complain terakhir yang ada di nama:

```
$nama = isset($_GET['nama']) ?  
mysqli_real_escape_string($conn, $_GET['nama'])  
: '';  
  
if (empty($nama)) {  
  
http_response_code(400);  
  
echo json_encode(["error" => "Parameter  
nama tidak ditemukan"]);  
  
exit; }  
  
h) Dashboard Survey
```



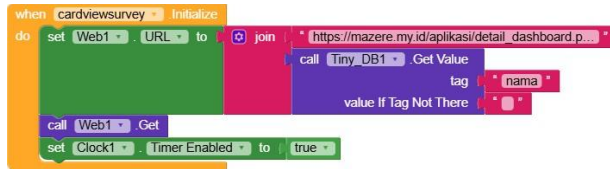
Gambar 29. *Home Survey Satisfaction*

Halaman *Home Survey Satisfaction* merupakan informasi utama yang dirancang untuk memudahkan pengguna dalam memantau aktivitas terkait laporan keluhan yang terakhir dibuat. Halaman ini menampilkan data *Work Order* terakhir yang dikerjakan/ di *request*. Halaman ini berfungsi sebagai halaman awal pengguna dalam memberikan penilaian terhadap kinerja untuk mengukur tingkat kepuasan pengguna terhadap layanan yang diberikan dalam penanganan keluhan.

Pada halaman ini terdapat *cardview* yang berisi data-data FPK yang dimana data tersebut akan muncul apabila status *Work Order* sudah mencapai *fully approved* (selesai dikerjakan dan memiliki persetujuan lengkap) sehingga *Work Order* belum bisa diberi penilaian apabila persetujuan yang diberikan belum lengkap. Pada komponen *cardview* terdapat data no FPK, tanggal, nama pelapor,

departement, tenancy dan status yang dimana apabila *cardview* tersebut ditekan maka akan menuju halaman *Survey Satisfaction* untuk memberikan penilaian kinerja *Engineer*.

Berikut blok program untuk menampilkan data work order terakhir sesuai dengan akun yang sedang login.



Gambar 30. Blok Program *Get Data Survey Satisfaction*

Berikut merupakan php untuk menampilkan data work order yang terakhir di input sesuai dengan akun yang sedang login:

```
$nama = isset($_GET['nama']) ?  
mysqli_real_escape_string($conn, $_GET['nama'])  
:"";
```

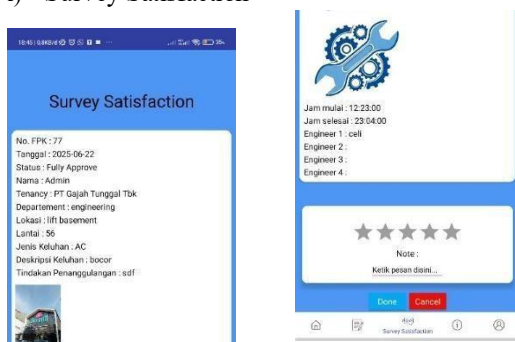
```
if (empty($nama)) {
```

```
http_response_code(400);
```

```
echo json_encode(["error" => "Parameter nama  
tidak ditemukan"]);
```

```
exit;} 
```

i) Survey Satisfaction

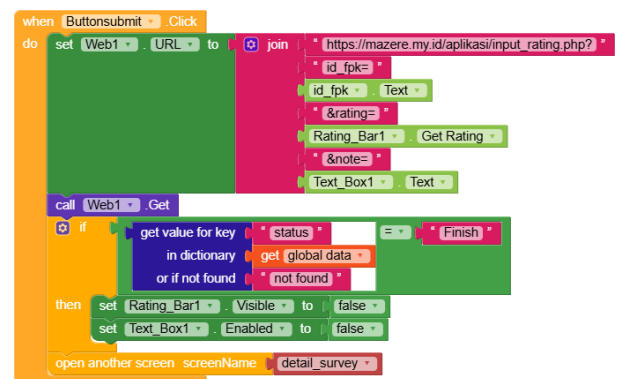


Gambar 31. *New Survey Satisfaction*

Halaman *New Survey Satisfaction* merupakan halaman untuk mengukur tingkat kepuasan pengguna terhadap layanan yang diberikan dalam penanganan keluhan. Ketika *cardview* di *Home Survey Satisfaction* ditekan, maka halaman aplikasi akan berpindah ke halaman *Survey Satisfaction*. Pada halaman *Survey Satisfaction* terdapat detail dari *Work Order* berupa no FPK, nama pelapor,

departement, lantai, lokasi, tenancy, status, jenis keluhan, deskripsi keluhan, foto keluhan, foto pengerjaan, foto selesai, tindakan penanggulangan, jam mulai, jam selesai pengerjaan, nama Engineer sehingga pengguna hanya cukup mengisi *rating* dan *note* untuk kinerja *Engineering*. Setelah mengisi *rating* dan *note*, pengguna diwajibkan untuk *submit* penilaian ke pihak *Engineering* dengan menekan tombol *submit*. Saat tombol *submit* ditekan maka kolom *rating* dan *note* di *database* akan ter-update.

Berikut merupakan program blok kodular untuk input survey satisfaction:



Gambar 32. Blok Program *Input Survey Satisfaction*

Dan berikut merupakan php untuk update data survey satisfaction:

```
// Gabungkan bagian SET dan buat query
```

```
$set_clause = implode(' ',  
$update_parts);
```

```
$sql = "UPDATE work_order SET $set_clause  
WHERE id_fpk = '$id_fpk'";
```

j) Detail Survey Satisfaction

Gambar 33. Detail Survey Satisfaction

Halaman *Detail Survey Satisfaction* merupakan halaman yang menampilkan informasi lengkap terkait *Work Order* sesuai dengan yang terakhir yang dibuat. Data diambil sesuai nama akun dan dari akun tersebut akan diambil data FPK yang terbesar. Data yang ditampilkan pada halaman ini diantaranya no FPK, nama pelapor, *departement*, lantai, lokasi, *tenancy*, no FPK, status, jenis keluhan, deskripsi keluhan, foto keluhan, foto pengerjaan, foto selesai, tindakan penanggulangan, jam mulai, jam selesai pengerjaan, nama *Engineer*, *rating* dan *note*.

Berikut merupakan blok program untuk tampil data fitur survey satisfaction yang dimana komponen clock berfungsi sebagai auto refresh web1:



Gambar 34. Blok Program Get Data Detail Survey

Dan berikut php untuk tampil data detail survey:

// Ambil data terakhir berdasarkan nama

```
$query = "SELECT * FROM work_order
WHERE nama = '$nama' ORDER BY
LENGTH(id_fpk) DESC, id_fpk DESC
```

```
LIMIT 1 ";
```

k) Halaman Instruksi



Gambar 35. Instruksi Aplikasi

Halaman *Instruksi* berfungsi sebagai panduan bagi pengguna dalam menggunakan aplikasi secara tepat dan efisien. Di halaman ini, dijelaskan langkah-langkah penggunaan setiap fitur utama, mulai dari proses *Login*, pengisian *Form Complain*, hingga cara melakukan penilaian pada layanan yang telah diberikan. Setiap instruksi disusun secara sistematis dan disertai dengan penjelasan yang mudah dipahami, sehingga memudahkan pengguna dari berbagai latar belakang untuk mengikuti alur penggunaan aplikasi. Beberapa bagian dalam halaman ini dilengkapi dengan gambar atau ikon penunjuk untuk memberikan ilustrasi visual yang memperjelas arahan. Dengan adanya halaman *Instruksi* dapat menghindari kesalahan *input* data, mempercepat pemahaman pengguna baru. Dengan demikian, halaman ini berperan sebagai komponen pendukung utama dalam menciptakan pengalaman pengguna yang baik dan efisien. Terdapat komponen web viewer yang berfungsi untuk menampilkan file pdf yang disimpan di google drive.

Berikut blok program halaman instruksi yang berfungsi untuk menuju ke file google drive :



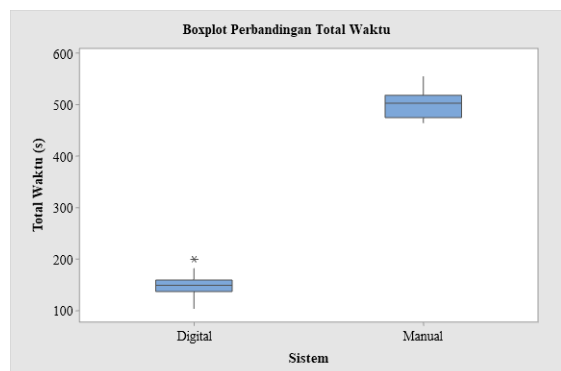
Gambar 36. Blok Program Instruksi

B. Pengambilan Data Potensial Sistem Manual dan Sistem Digital

Data potensial sistem manual dan digital diolah menggunakan metode statistik deskriptif dengan menggunakan *software Minitab*, dengan tujuan untuk mengetahui rata-rata waktu sistem manual dan sistem digital, standar deviasi, serta rentang waktu tercepat dan terlama yang terjadi pada tiap sistem. Hasil pengolahan data menunjukkan bahwa rata-rata total waktu sistem manual adalah sebesar 501.13 detik dengan standar deviasi 25.56 detik. Sedangkan, sistem digital memiliki rata-rata total waktu sebesar 148.50 detik dengan standar deviasi 20.10 detik. Standar deviasi yang jauh lebih rendah pada sistem digital menunjukkan bahwa proses berjalan lebih konsisten dan stabil dibandingkan sistem manual.

Tabel 3. Statistik Deskriptif Total Waktu Proses

Vari able	N	Me an	St De v	Mini mum	Med ian	Maxi mum
Siste m Man ual	30	501.13	25.56	465.00	503.00	554.00
Siste m digit al	30	148.50	20.10	105.00	150.00	200.00

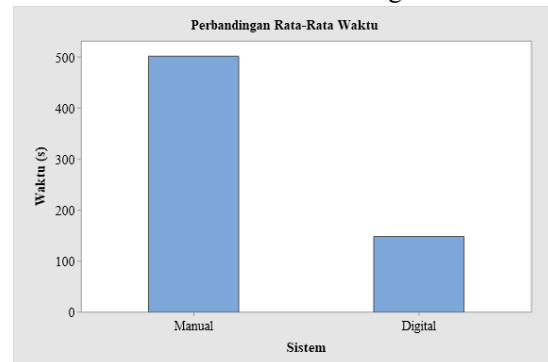


Gambar 37. Boxplot Perbandingan Total Waktu Proses

C. Perbandingan Efisiensi Sistem

Setelah dilakukan pengambilan dan pengolahan data terhadap waktu proses sistem manual dan digital, dilakukan perhitungan efisiensi guna mengetahui sejauh mana sistem digital mampu meningkatkan kinerja proses. Pada sistem manual dan sistem digital, analisis ini didasarkan pada rata-

rata total waktu yang dibutuhkan untuk menyelesaikan satu alur kerja dari awal hingga selesai. Dari hasil analisis statistik deskriptif, menunjukkan bahwa adanya penurunan waktu proses yang cukup signifikan setelah penerapan sistem digital. Untuk memperjelas selisih waktu antar sistem, Pada Gambar 87 menampilkan diagram perbandingan rata-rata total waktu proses antara sistem manual dan sistem digital.



Gambar 38. Diagram Perbandingan Rata-Rata Total Waktu

Nilai efisiensi dihitung menggunakan rumus berikut ini:

$$\begin{aligned}
 \text{Efisiensi (\%)} &= \frac{\text{Waktu Manual} - \text{Waktu digital}}{\text{Waktu Manual}} \\
 &\quad \times 100\% \\
 &= \frac{501,13 \text{ detik} - 148,50 \text{ detik}}{501,13 \text{ detik}} \times 100\% \\
 &= 70,36\%
 \end{aligned}$$

Dengan nilai efisiensi sebesar 70,36 % dapat disimpulkan bahwa sistem digital mampu mempercepat proses kerja secara signifikan dibandingkan sistem manual. Selain itu, sistem digital juga mengurangi tahapan proses kerja sistem manual yang sebelumnya memerlukan intervensi langsung dari Admin maupun *Engineer*, sehingga meminimalisir kemungkinan keterlambatan, kesalahan pencatatan, dan duplikasi data. Hasil ini mendukung tujuan utama dari pengembangan sistem digital, yaitu untuk meningkatkan kecepatan, efisiensi, dan konsistensi proses kerja *complain*.

D. Perbandingan Biaya Operasional Sistem

Selain efisiensi waktu, aspek biaya juga menjadi pertimbangan penting dalam evaluasi sistem digitalisasi. Perbandingan biaya operasional antara

sistem manual dan sistem digital dilakukan untuk mengetahui sejauh mana sistem digital mampu menekan pengeluaran rutin yang selama ini dibutuhkan dalam proses pengelolaan FPK. Berdasarkan data yang diperoleh, sistem manual memerlukan beberapa komponen biaya, seperti pembelian buku FPK yang terdiri dari tiga rangkap dalam satu kertas untuk setiap pengajuan *complain*. Dengan total kebutuhan FPK sebanyak 2.732 request per tahun, dan satu buku berisi 150 lembar, maka kebutuhan buku FPK per tahun mencapai 19 buku. Jika dikalikan dengan harga satuan, maka total biaya operasional sistem manual sebesar Rp 1.366.000 per tahun.

Tabel 4. Biaya Operasional Sistem Manual

Komponen	Satuan	QTY	Biaya Satuan	Total
Kertas per FPK (3 rangkap)	Request	2732	Rp 500	Rp 1,366,000
Total Biaya Manual per Tahun			Rp 1,366,000	

Sementara itu, sistem digital hanya membutuhkan biaya untuk layanan hosting *database* sebesar Rp 20.000 per bulan. Jika dikalikan selama satu tahun, maka total biaya operasional sistem digital sebesar Rp 240.000 per tahun. Seluruh proses pencatatan, pengiriman, dan arsip dilakukan secara daring, sehingga tidak lagi membutuhkan biaya kertas dan distribusi fisik.

Tabel 5. Biaya Operasional Sistem Digital

Komponen	Satuan	QTY	Biaya Satuan	Total
Hosting Aplikasi	Bulan	12	Rp 20,000	Rp 240,000
Total Biaya Digital per Tahun			Rp 240,000	

Efisiensi biaya dihitung menggunakan selisih antara kedua sistem, dan diperoleh hasil sebesar:

Efisiensi (%)

$$= \frac{\text{Biaya Manual} - \text{Biaya digital}}{\text{Biaya Manual}} \times 100\%$$

$$= \frac{1.366.000 - 240.000}{1.366.000} \times 100\%$$

$$= 82,43 \%$$

Dengan efisiensi biaya sebesar 82,43 %, sistem digital terbukti jauh lebih hemat dan efisien dalam penggunaan anggaran operasional. Selain menekan pengeluaran rutin, sistem digital juga mengurangi ketergantungan terhadap material cetak seperti kertas dan proses administratif yang berulang, sehingga lebih ramah lingkungan dan mudah diakses.

E. Pengujian Sistem

Tabel 6. Blackbox Testing Sistem Aplikasi

No	Halaman Uji	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Login	Klik tombol Login setelah memasukkan Username dan password sesuai dengan role	Menampilkan halaman Dashboard utama	OK
2	Profile	Klik button navigation Profile	Menampilkan data profil User sesuai dengan akun yang sedang Login.	OK
3	Dashboard History	Klik bottom navigation History	Menampilkan aplikasi terdapat listview dashboard	OK

IV. KESIMPULAN

Perancangan dan pembangunan sistem Aplikasi Tenant berbasis Android telah berhasil dilakukan menggunakan platform Kodular, dengan menggunakan rancangan antarmuka pengguna yang dibuat menggunakan *tools* draw.io untuk memastikan tampilan yang intuitif dan mudah digunakan. Aplikasi ini dikembangkan dengan tujuan untuk memberikan kemudahan bagi para pengguna, khususnya bagi tenant dalam mengakses

informasi yang berkaitan dengan data tenant secara real time. Selain itu, aplikasi ini juga dirancang untuk memfasilitasi proses pengajuan permintaan kerja (job order) kepada pihak Engineering secara cepat, praktis dan efisien tanpa harus melalui proses manual. Melalui integrasi fitur-fitur yang sesuai dengan kebutuhan pengguna, aplikasi ini diharapkan dapat meningkatkan efektivitas komunikasi antara tenant dan pihak Engineering serta mendukung digitalisasi proses manajemen gedung.

DAFTAR PUSTAKA

- [1] N. Elyasi, A. Bellini, and N. J. Klungseth, "Digital transformation in facility management: An analysis of the challenges and benefits of implementing digital twins in the use phase of a building," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1176, no. 1, 2023, doi: 10.1088/1755-1315/1176/1/012001.
- [2] S. Eko, "Perancangan Aplikasi Pengenalan Budaya Nusantara Berbasis Android Dengan Metode Rad," *J. Ilmu Komput. JIK*, vol. 5, no. 01, pp. 30–39, 2022.
- [3] F. N. Hasanah, *Buku Ajar Rekayasa Perangkat Lunak*. 2020. doi: 10.21070/2020/978-623-6833-89-6.
- [4] Siska Narulita, Ahmad Nugroho, and M. Zakki Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS)," *Bridg. J. Publ. Sist. Inf. dan Telekomun.*, vol. 2, no. 3, pp. 244–256, 2024, doi: 10.62951/bridge.v2i3.174.
- [5] R. Rohmanto and T. Setiawan, "Perbandingan Efektivitas Sistem Pembelajaran Luring dan Daring Menggunakan Metode Use case dan Sequence Diagram," *Intern. (Information Syst. Journal)*, vol. 5, no. 1, pp. 53–62, 2022, doi: 10.32627/internal.v5i1.506.
- [6] M. Ahmadar, P. Perwito, and C. Taufik, "PERANCANGAN SISTEM INFORMASI PENJUALAN BERBASIS WEB PADA RAHAYU PHOTO COPY DENGAN DATABASE MySQL," *Dharmakarya*, vol. 10, no. 4, p. 284, 2021, doi: 10.24198/dharmakarya.v10i4.35873.
- [7] E. Trivaika and M. A. Senubekti, "Perancangan Aplikasi Pengelola Keuangan Pribadi Berbasis Android," *Nuansa Inform.*, vol. 16, no. 1, pp. 33–40, 2022, doi: 10.25134/nuansa.v16i1.4670.
- [8] T. A. Rismayanti, N. Anriani, and S. Sukirwan, "Pengembangan E-Modul Berbantu Kodular pada Smartphone untuk Meningkatkan Kemampuan Berpikir Kritis Matematis Siswa SMP," *J. Cendekia J. Pendidik. Mat.*, vol. 6, no. 1, pp. 859–873, 2022, doi: 10.31004/cendekia.v6i1.1286.
- [9] U. Kholifah and N. Imansari, "Pelatihan Membangun Aplikasi Mobile Menggunakan Kodular Untuk Siswa Smpn 1 Selorejo," *Abdimas Galuh*, vol. 4, no. 1, p. 549, 2022, doi: 10.25157/ag.v4i1.7259.
- [10] N. K. Hendriawan, B. Basrowi, and K. Rahmadani, "Pengembangan Media Pembelajaran Aplikasi Kodular pada Mata Pelajaran Teknologi Informasi dan Komunikasi Kelas X di SMK Pasudan 1 Kota Serang," *JiIP - J. Ilm. Ilmu Pendidik.*, vol. 6, no. 12, pp. 10491–10495, 2023, doi: 10.54371/jiip.v6i12.2927.
- [11] S. Salsabila, N. Anriani, and C. A. H. F. Santosa, "Pengembangan E-Modul Pada Android Menggunakan Kodular Untuk Meningkatkan Kemampuan Representasi Matematis Siswa," *Teorema Teor. dan Ris. Mat.*, vol. 8, no. 1, p. 1, 2023, doi: 10.25157/teorema.v8i1.8704.
- [12] R. Hermiati, A. Asnawati, and I. Kanedi, "Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql," *J. Media Infotama*, vol. 17, no. 1, pp. 54–66, 2021, doi: 10.37676/jmi.v17i1.1317.
- [13] A. A. Nozari, "Interface Data Sistem ERP SAP Dan Aplikasi Android Di Server Hosting Menggunakan Flat File (Studi Kasus: Aplikasi Mobile Populasi Sapi Pada PT. Great Giant Livestock)," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 158–166, 2020, doi: 10.33365/jatika.v1i2.362.
- [14] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, and Q. A. Giansyah, "Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web," *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 1–16, 2023.
- [15] E. Novalia and A. Voutama, "Black Box Testing dengan Teknik Equivalence

- Partitions Pada Aplikasi Android M-Magazine Mading Sekolah,” *Syntax J. Inform.*, vol. 11, no. 01, pp. 23–35, 2022, doi: 10.35706/syji.v11i01.6413.
- [16] N. M. D. Febriyanti, A. . K. Oka Sudana, and I. N. Piarsa, “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen,” *JITTER J. Ilm. Teknol. dan Komput.*, vol. 2, no. 3, p. 535, 2021, doi: 10.24843/jrti.2021.v02.i03.p12.
- [17] S. Shidqi, D. A. Sulistyono, and F. A. Ahda, “Pembuatan Infrastruktur Database Menggunakan Metode Replikasi Untuk Pelanggan Jagoan Hosting,” *J. Ilm. Teknol. Inf. Asia*, vol. 16, no. 1, pp. 65–74, 2022, doi: 10.32815/jitika.v16i1.702.
- [18] L. M. Rohmah, “Studi Literatur: Media Pembelajaran Melalui Model Pembelajaran Discovery Learning dalam meningkatkan hasil Belajar Siswa pada pembelajaran Biologi Tahun 2022/2023,” *Skripsi Univ. Islam Negeri Kiai Achmad Siddiq Jember*, 2023.
- [19] O. Eko Anggraini, A. Prio Agus Santoso, E. Chotidjah Suhatmi, and U. Hanifah, “Analisis Yuridis Pengupahan Tenaga Kesehatan Kontrak Di Rumah Sakit,” *J. Bisnis Manaj. dan Akunt.*, vol. 1, no. 2, pp. 78–84, 2021, doi: 10.47701/bismak.v1i2.1207.
- [20] P. Utomo, N. Asvio, and F. Prayogi, “Metode Penelitian Tindakan Kelas (PTK): Panduan Praktis untuk Guru dan Mahasiswa di Institusi Pendidikan,” *Pubmedia J. Penelit. Tindakan Kelas Indones.*, vol. 1, no. 4, p. 19, 2024, doi: 10.47134/ptk.v1i4.821.