p–ISSN: 2723 - 6609 e-ISSN: 2745-5254
Vol. 5, No. 1 January 2024 http://jist.publikasiindonesia.id/

ANDROID-BASED E-INVENTORY SYSTEM (CASE STUDY: DPRD KABUPATEN BERAU)

Muhammad Daffa Firmansyah[1*], Wahyu Sri Utami[2]
Universitas Teknologi Yogyakarta, Indonesia
Email : daffafirmansyah0@gmail.com[1*], wahyu.utami@staff.uty.ac.id[2]

*Correspondence

Keywords: Android, Inventory, Database, Documentation, E-government.

ABSTRACT

Technology is a rapidly evolving field in recent times, capable of solving various problems across different sectors. One problem addressed in the government sector is the emergence of e-government, a movement where technology is utilized to carry out government activities more effectively. One form of implementing e-government is through the application of e-inventory. The purpose of developing this application is to provide a solution to issues related to documenting state-owned goods, such as errors in manual data recording, lack of real-time data updates for the goods, and difficulties in accessing data during inspections conducted by the Inspectorate or the Financial Audit Agency (BPK). These problems often occur in government offices with a large inventory, such as the Regional People's Representative Council of Berau (DPRD KAB.Berau), which still faces challenges in recording inventory items. Therefore, an Android-based e-inventory application has been developed to facilitate activities such as updating inventory data, documenting goods, and simplifying data retrieval.

Introduction

The rapid development of information technology can solve many problems in various industries (Tarantang, Awwaliyah, Astuti, & Munawaroh, 2019). With Presidential Instruction (Inpres) Number 3 of 2003, the government realized that using communication and information technology in government processes (e-government) can increase the efficiency, effectiveness, transparency, and accountability of government administration (Syamsuar & Reflianto, 2019). The government sector cannot be separated from technological advances, where the government must be able to use information and communication technology to implement government systems (Iswandari, 2021).

However, there is a problem in implementing this e-government where the inventory documentation system still exists, which is only recorded on paper even though the latest data has been changed and has begun to be stored in an Excel file. This problem must be resolved immediately because state property management must know precisely the assets owned (Hudiya, Puspita, Kawigraha, & Hapid, 2021).

Another problem is the slow update of existing stock items. This problem can cause data discrepancies because the goods data must be audited manually. This can cause data differences in reports and different quantities of items (Kasemin, 2016).

The above problems can lead to new problems where when an inspection of office goods is held. During the inspection, supervisors need the latest data as soon as possible.

Employees must first check the stock of goods and update the data on Excel, which results in slow updating of the required data (Manoharan & Ingrams, 2018).

Technology that can solve the above problems. One of the technologies that can overcome the above problems is a mobile-based e-inventory application (Lindo & Tukino, 2023). Mobile applications are chosen because they can be connected to cloud servers, which will help store data anywhere the user wants to use the application as long as the user is connected to the internet network (Hay, Safaah, & Ramadhan, 2019). In addition, the purpose of this application is to facilitate the process of updating the stock inventory of consumables, make it easier for users to document goods, and facilitate data retrieval to increase work effectiveness (Sinaga & Rahayu, 2018). The development of this application aims to be a development for the development of e-government applications in the Berau Regency DPRD related to inventory for consumables.

Consumable goods are those included in state wealth that are consumable for official purposes or for a period of use that is less than one year (Juansyah, 2018). An example of consumables is stationary; the purpose of consumables in this study is because of the short period of use of these goods, which requires that the goods must constantly be audited.

This application aims to improve the recording and administration services of the use of consumables in the fields in the DPRD Secretariat and realize the Development of an integrated consumables recording application in the Berau Regency DPRD through the Android Application.

Research Methods

Data obtained

The data obtained is a file containing an inventory list of consumables recorded between 2020 and 2021, where the data contains the date of the item update, the item's name, and the item's quantity and unit. The sample of the file in question can be seen in the following figure.

Figure 1 Goods request letter

Data collection procedure

Data is obtained by directly interviewing staff from related agencies and requesting copies of several documents. Data collection is carried out online and in stages with vulnerable data collection between October 16, 2022, and April 25, 2023

Business rules

The system to be analyzed is a system that will later be used to monitor the movement of inventory goods. Hopefully, this system can meet the needs and overcome existing problems.

Stages of Research

At this stage, the author explains what activities will be carried out in this research activity, as the activities can be seen in the following picture.

Figure 2 Research stage flow

Results and Discussion

Splash screen Page implementation

The initial view when the application is run before heading to the login page. A splash screen page is displayed so that users understand the application to be used. Here is the splash screen page of the system application.

Figure 3 Splash Screen Display

Login Page Implementation

The login page displays the login form when the user already has an account and enters the email and password created in this application. However, if the user forgets the password can click forgot password above the login button.

Figure 4 Screan Login Display

Then, to authenticate to Firebase authentication, you can use the following code:

```
auth.signInWithEmailAndPassword(email,password).addOnCompleteLis
tener (this){
            if(it.isSuccessful){
            val user = auth.currentUser
                getuserrole(user?.uid)
            }else
                Toast.makeText(this,"auth
failed",Toast.LENGTH_SHORT).show()
        }
```

Menu Page Implementation

This page is displayed after the user successfully enters the system; this page displays request data for admins and leaders and several buttons for the user; the menu page can be seen in the picture.

Figure 5 Menu Display

The code to retrieve data is as follows:

```
viewPager2.adapter=adapter

        tasklist = mutableListOf()
        db.addValueEventListener(object : ValueEventListener{
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()){
                    for (h in snapshot.children){
                        val                 data                 =
h.getValue(dataPermintaan::class.java)
                        if (data != null){
                            tasklist.add(data)
                                }
                    }
                    val                  adapter                 =
taskadapter(applicationContext,R.layout.taskcard,tasklist)
                    lv.adapter = adapter
                }
```

Inventory page implementation

This page will display data from existing inventory, and on this page, admins can add inventory data that is not yet on this page. Also, admins can update data and delete inventory data. The display of the inventory page can be seen in the following figure.

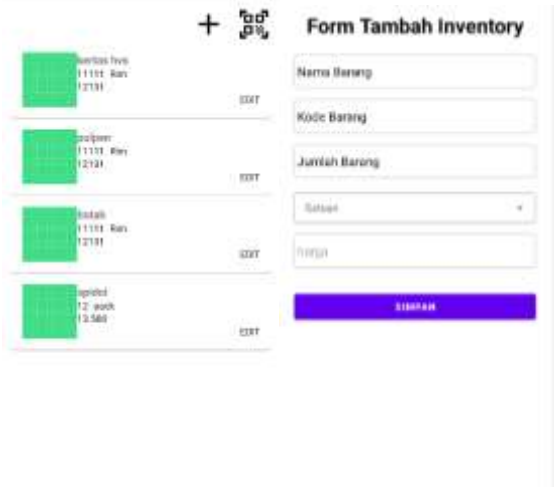

Figure 6 Inventory Display

The algorithm for adding data can be seen in the following code:

```
                val databaseReference =
        FirebaseDatabase.getInstance().reference
val datainventory = databaseReference.child("Inventory")

      dataref.setValue(data).addOnSuccessListener {
    Toast.makeText(this,"data berhasil di tambahkan",
                Toast.LENGTH_SHORT).show()
                 }.addOnFailureListener{
      Toast.makeText(this,"gagal menambahkan data ",
                Toast.LENGTH_SHORT ).show()
```

Implementation of the Add User page

On this page, the admin can add users who can use the e-inventory system; here is the appearance of the add user page:

Figure 7 Add user view.

The algorithm for adding users can be seen in the following code:

```
            auth.createUserWithEmailAndPassword(email,
             password).addOnCompleteListener { task ->
                            if (task.isSuccessful) {
                         val firebaseuser = task.result?.user
                                uid = firebaseuser?.uid
                       Toast.makeText(this, "User Berhasil di
          tambahkan, $uid", Toast.LENGTH_SHORT).show()
                                     } else {
                        Toast.makeText(this, "user gagal di
             tambahkan", Toast.LENGTH_SHORT).show()
                                        }
```

Implementation of the Submit a Request Page

On this page, the User can add a request, which can later be processed by the admin of the request page, which can be seen in the picture.

Figure 8 Request View

The request code can be seen in the following code:

```
   dataref.setValue(data).addOnSuccessListener {
                    Toast.makeText(this,"data       berhasil       di
tambahkan",Toast.LENGTH_SHORT).show()
                }.addOnFailureListener{
                    Toast.makeText(this,"data  gagal  di  tambahkan",
Toast.LENGTH_SHORT).show()
```

System Testing

Table 1 System Testing

NO	Testing	Picture	Information	Conclusion
1	Splash Screen		Display the app's start page and display an animated app logo	Succeed
2	Login		Display the Login form.	Succeed
3	Forgot password		Display the forgot password form and send a recovery email	Succeed

4	Profile		Display the details of the logged-in user data	succeed

Admin System Testing

Table 2
System Admin Testing

NO	Testing	Information	Conclusion
1	Main menu	It can display request data, announcement banner, and admin data, and there are 2 add user and stock buttons that, when clicked, can move to another activity.	Succeed
2	Add user	Can add users who can access the system	Succeed
3	stock	Can display inventory stock, update inventory data, delete inventory data, and add inventory data	Succeed

User System Testing

Table 3
User System Testing

NO	Testing	Information	Conclusion
1	Main menu	It can display login user data, an announcement banner, and four buttons to switch activities.	Succeed
2	Inventors	Display inventory data of available goods	Succeed
3	Request history	Surface user request history	succeed
4	Demand	Bring up a running request transaction	Succeed
5	Make a request	Bring up the request form, and when the submit button is pressed, save data in Firebase.	Succeed

Lead system testing

Table 4
Lead System Testing

No	Testing	Information	conclusion
1	Main Menu	It displays request data that the leader can approve or reject, several buttons to switch activities, and an announcement banner.	Succeed
2	Stock	Can Display Stock Data of Existing Goods	Succeed
3	History	Displays data from requests that have been approved or rejected	Succeed

Conclusion

It can run well based on research that has been made and tested. The e-inventory system developed using the Android platform can facilitate the process of inventory management and increase efficiency in decision-making. The e-inventory application also allows users from various locations easy access, making monitoring inventory in real-time more accessible. Therefore, adding features and improvements related to existing limitations is necessary for further development.

Bibliography

Hay's, Riyan Naufal, Safaah, Eva, & Ramadhan, Agi Febrian. (2019). Sistem Inventory Suku Cadang Berbasis Client-Server Pada Cv. Sarana Mandiri Motor. *Jurnal Dinamika Informatika*, *8*(2), 1–11.

Hudiya, Noval, Puspita, Angella Natalia Ghea, Kawigraha, Adji, & Hapid, Abdul. (2021). Pengembangan Aplikasi E-Inventory Barang Inventaris Negera di PTPSM-BPPT. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(4), 823–830. https://doi.org/10.25126/jtiik.2021844504

Iswandari, Bunga Asoka. (2021). Jaminan Atas Pemenuhan Hak Keamanan Data Pribadi Dalam Penyelenggaraan E-Government Guna Mewujudkan Good Governance. *Jurnal Hukum Ius Quia Iustum*, *28*(1), 115–138.

Juansyah, Juansyah. (2018). E-INVENTORY PERUSAHAAN DAERAH AIR MINUM (PDAM) TIRTA RANDIK. *Jurnal TIPS: Jurnal Teknologi Informasi Dan Komputer Politeknik Sekayu*, *9*(2), 54–61.

Kasemin, H. Kasiyanto. (2016). *Agresi Perkembangan Teknologi Informasi*. Prenada Media.

Lindo, Junius, & Tukino, Tukino. (2023). Rancang Bangun E-Inventory Spare Part Kapal Berbasis Codeigniter pada PT Pelayaran Nasional Sandico Ocean Line Batam. *Computer Based Information System Journal*, *11*(2), 16–27.

Manoharan, Aroon P., & Ingrams, Alex. (2018). Conceptualizing e-government from local government perspectives. *State and Local Government Review*, *50*(1), 56–66.

Sinaga, Tantri Hidayati, & Rahayu, Eka. (2018). Analisis dan Perancangan e-Inventory Instrumen Perkuliahan pada STT Harapan Medan. *Query: Journal of Information Systems*, *2*(2).

Syamsuar, Syamsuar, & Reflianto, Reflianto. (2019). Pendidikan dan tantangan pembelajaran berbasis teknologi informasi di era revolusi industri 4.0. *E-Tech: Jurnal Ilmiah Teknologi Pendidikan*, *6*(2). https://doi.org/10.24036/et.v2i2.101343

Tarantang, Jefry, Awwaliyah, Annisa, Astuti, Maulidia, & Munawaroh, Meidinah. (2019). Perkembangan sistem pembayaran digital pada era revolusi industri 4.0 di indonesia. *Jurnal Al-Qardh*, *4*(1), 60–75.