

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam melaksanakan kerja magang di Badan Pengelolaan Aset Daerah DKI Jakarta, penulis memegang jabatan sebagai *app developer* di divisi Data dan Informasi, dengan pengawasan dari Bapak Suropto selaku Kepala Sub Bidang Data & Informasi.

3.2 Tugas yang Dilaksanakan

Realisasi kerja magang ditunjukkan pada Tabel 3.1

Tabel 3.1 Realisasi Kerja Magang

Minggu	Kegiatan
1	Latihan dasar <i>android studio</i>
2	Perencanaan struktur aplikasi menurut permintaan supervisi
3	Implementasi <i>Scan QR Code</i> ke aplikasi dan mengatur ulang tampilan depan aplikasi.
4	Memperbaiki <i>error</i> fitur <i>Scan QR Code</i> , dan merencanakan fitur kamera untuk mengambil gambar
5	Merombak ulang aplikasi untuk mengganti <i>library scan QR Code</i> menjadi jenis yang berbeda
6	Implementasi fitur kamera ke aplikasi, yang dapat ditunjukkan tampilan depan aplikasi dan merencanakan fitur tambahan
7	Memperbaiki <i>error</i> dan menambahkan fitur memilih gambar yang dipilih, dapat diperoleh dari kamera langsung atau mengambil dari <i>gallery</i>
8	Memperbaiki <i>bug</i> yang ditemukan

3.3 Uraian Pelaksanaan Kerja Magang

Pelaksanaan kerja magang dibagi ke dalam 4 bagian, yakni, analisa kebutuhan, perancangan aplikasi, proses pelaksanaan, kendala dalam proses pengerjaan, solusi atas kendala yang dihadapi.

3.3.1. Analisa Kebutuhan

Pada rancang bangun aplikasi sensus barang modul *QR Code scanner* terdapat kebutuhan yang harus dipenuhi, yaitu:

1. Hasil Scan QR Code yang berupa ID Barang dapat ditampilkan di halaman depan
2. Dapat menentukan kondisi dari barang
3. Sistem dapat *upload* gambar yang ditampilkan di layar depan
4. Sistem hanya akan dapat *men-submit* jika semua kondisi terpenuhi

3.3.2. Proses Pelaksanaan

Pembuatan aplikasi sensus barang modul *QR Code scanner* menggunakan laptop pribadi penulis yang memiliki spesifikasi sebagai berikut:

1. Processor Intel(R) Core i3
2. VGA NVIDIA GeForce 720M
3. RAM 4 GB

Perangkat lunak yang digunakan adalah sebagai berikut :

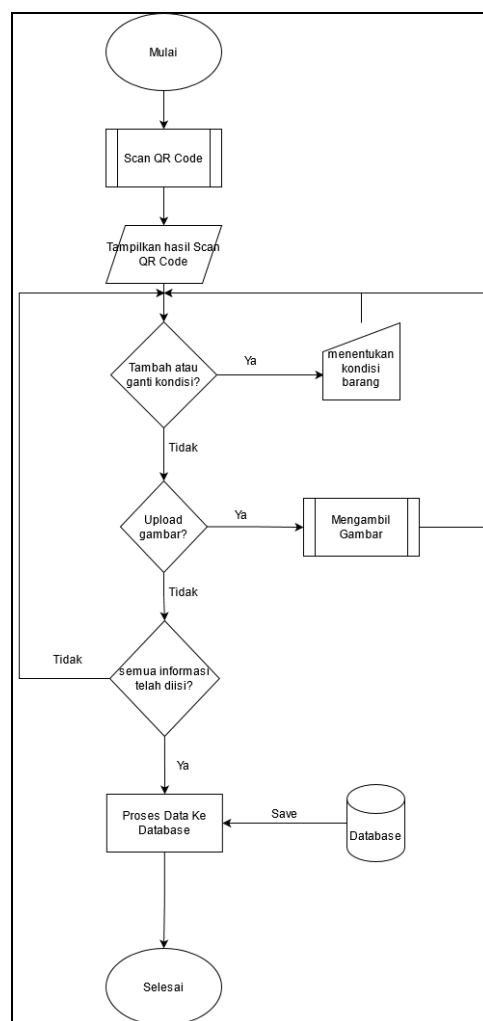
1. Android Studio menggunakan bahasa pemrograman Java.
2. OS Windows 8

Pengerjaan perancangan aplikasi sensus barang modul *QR Code scanner* dibagi menjadi 2 tahap, yaitu perancangan dan implementasi.

A. Perancangan

Tahap perancangan dalam pembuatan aplikasi sensus barang modul *QR Code scanner* dibantu dengan adanya *flowchart*.

A.1. Flowchart Perancangan Alur Penggunaan Aplikasi Sensus Barang

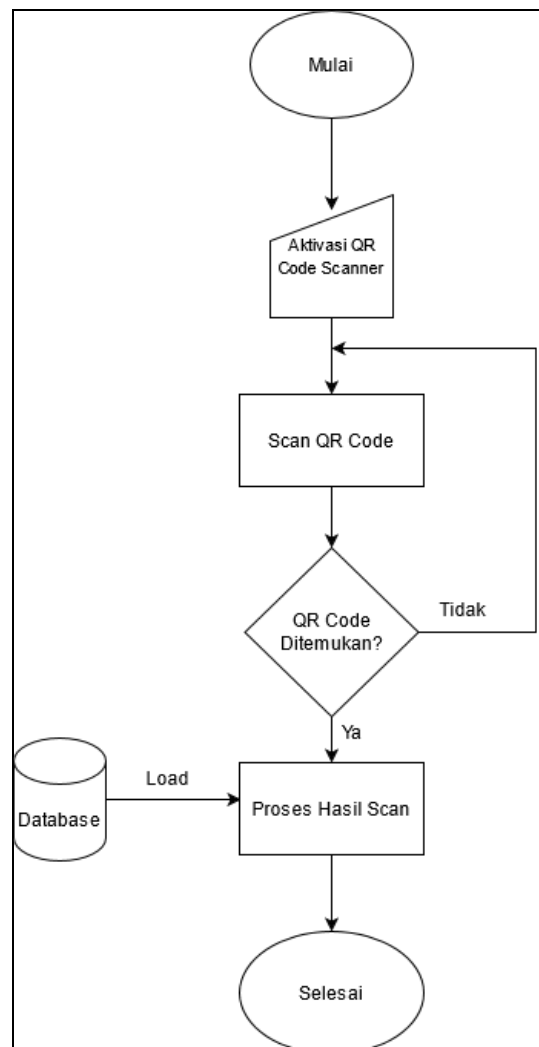


Gambar 3.1 *Flowchart* Perancangan Alur Penggunaan Aplikasi Sensus Barang

Gambar 3.1 berisikan *Flowchart* menampilkan rancangan alur penggunaan aplikasi sensus barang. *Flowchart* ini menampilkan menampilkan semua menu

yang diperlukan untuk memenuhi permintaan yang telah diberikan. Saat aplikasi dijalankan, user akan diminta untuk mengisi data yang diperlukan seperti, *Scan QR Code*, input kondisi barang, dan *Upload Gambar*. User hanya dapat menyimpan gambar ketika semua data telah dipenhi. Untuk kondisi barang sendiri, disediakan beberapa kriteria yang menunjukkan kondisi barang yang akan disimpan.

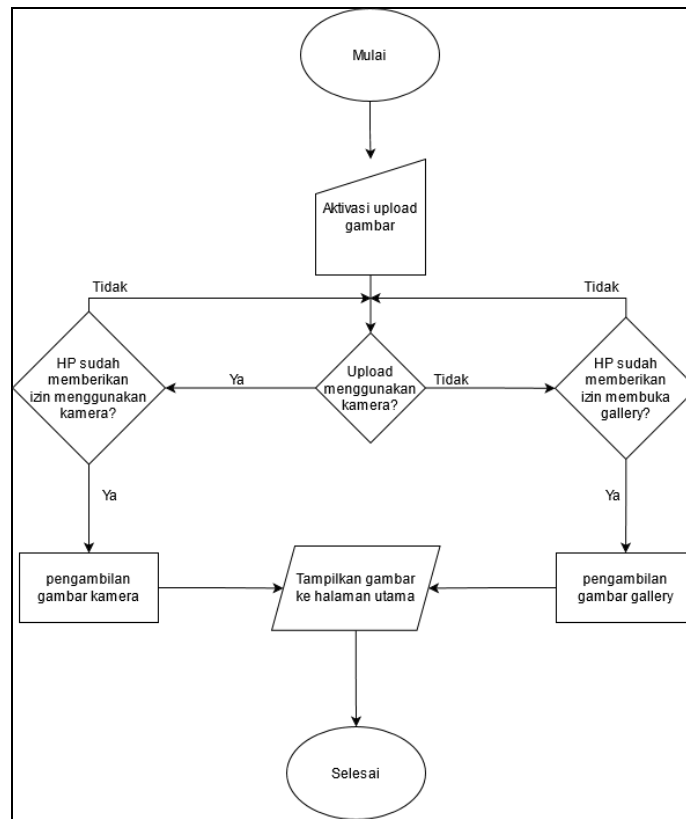
A.1.2 Flowchart Scan QR Code Aplikasi Sensus Barang



Gambar 3.2 *Flowchart* Proses *Scan QR Code* Aplikasi Sensus Barang

Gambar 3.2 Berisikan *Flowchart* proses *Scan QR Code* aplikasi sensus barang. Setelah mengaktifkan fitur *Scan QR Code* sebuah kamera akan dinyalakan, yang akan dipakai untuk melakukan aktifitas *scan*. Kamera tersebut akan berjalan sampai *user* mematikan aplikasi atau sebuah *QR Code* berhasil ter-*scan*. Hasil dari *scan* tersebut akan dikirimkan ke sebuah database, yang akan menampilkan data yang sebelumnya tersimpan. Jika belum ada data yang tersimpan, aplikasi akan menampilkan data yang kosong.

A.1.3 *Flowchart Upload Gambar Aplikasi Sensus Barang*



Gambar 3.3 *Flowchart* Proses *Upload* Gambar Aplikasi Sensus Barang

Gambar 3.3 berisikan *Flowchart* Proses *Upload* Gambar Aplikasi Sensus Barang. Saat user memilih untuk meng-*upload* sebuah gambar, terdapat 2 pilihan

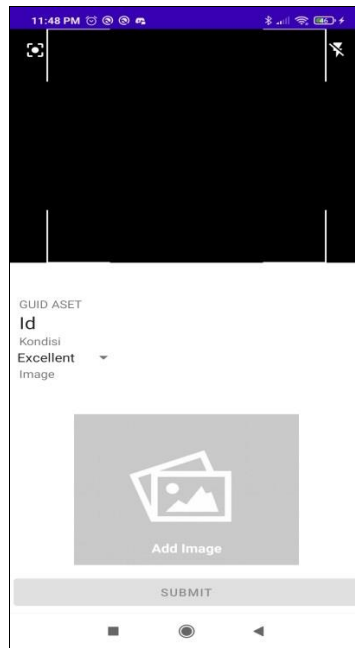
yang dapat dipilih yaitu, mengambil gambar langsung dari kamera dan mengambil gambar dari *gallery*. Jika memilih mengambil gambar langsung dari kamera, jika pertama kali dipakai akan muncul *pop-up* untuk mengkonfirmasi, bahwa apakah aplikasi dapat mendapatkan akses menggunakan kamera atau tidak, lalu user mengambil gambar dengan menggunakan kamera tersebut, jika gambar sudah mengkonfirmasi gambar yang diambil, hasil gambar akan dikirimkan langsung ke halaman utama. Jika user memilih untuk menggunakan gambar dari *gallery*, setelah memilih gambar yang akan digunakan, gambar tersebut akan langsung ditampilkan ke halaman utama.

B. Implementasi

Untuk tahap implementasi akan dibagi menjadi beberapa bagian yaitu, tampilan depan, xml, Java dan pengujian.

B.1 Tampilan Depan

Bagian tampilan depan ini berfungsi untuk menunjukkan *Interface* aplikasi sensus barang modul *QR Code Scanner* menggunakan *Screenshot*.



Gambar 3.4 Tampilan Depan Aplikasi Sensus Barang Modul QR Code Scanner

Gambar 3.4 menunjukkan gambar dari halaman depan aplikasi. Untuk mengaktifkan mode *scan QR Code*, user harus menekan layar di area yang hitam, area tersebut langsung menyalakan kamera dan dapat langsung *scan QR Code*. Lalu untuk memilih kondisi sebuah barang disediakan *spinner* yang terdapat beberapa pilihan untuk menentukan kondisi barang. Untuk meng-*Upload* sebuah gambar, user menekan gambar *add image* untuk memunculkan 2 pilihan, mengambil gambar dan upload gambar dari *gallery*. Untuk tombol *submit* akan di non-aktifkan selama semua data belum terpenuhi.

B.2 XML

Untuk bagian XML akan menunjukkan potongan kode pemrograman dalam file XML pada aplikasi sensus barang model *QR Code scanner*, yang dibantu dengan menggunakan *screenshot*.

```

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.4" />

```

Gambar 3.5 Bagian Kode Konfigurasi *Layout* Aplikasi

Gambar 3.5 menunjukkan konfigurasi dari *layout* aplikasi sensus barang modul *QR Code scanner*. Jenis *layout* yang digunakan adalah *constraint*, dengan tambahan sebuah *guideline* yang mempunyai id bernama *guideline* yang digunakan untuk membentangkan memberikan garis pembatas untuk *QR Code scanner* sehingga tidak merusak orientasi dari menu lainnya.

```

<com.budiyev.android.codescanner.CodeScannerView
    android:id="@+id/scanner_view"
    android:layout_width="@dp"
    android:layout_height="@dp"
    app:frameAspectRatioHeight="1"
    app:frameAspectRatioWidth="1"
    app:frameColor="@android:color/white"
    app:frameCornersRadius="0dp"
    app:frameCornersSize="70dp"
    app:frameSize="1"
    app:frameThickness="2dp"
    app:layout_constraintBottom_toTopOf="@+id/guideline"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:maskColor="#77000000" />

```

Gambar 3.6 Bagian Kode *Layout QR Code Scanner* Aplikasi

Gambar 3.5 menunjukkan potongan kode dari *layout QR Code scanner*. Fitur *QR Code scanner* akan diambil dari sebuah library yang bernama *codescanner* dan menggunakan *CodeScannerView* untuk menampilkan *layout*

kamera *QR Code scanner*. *Layout* ini memiliki sebuah id yang bernama *scanner_view* yang diletakkan diatas id *guideline*.

```
<ImageView
    android:id="@+id/imgItem"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginBottom="12dp"
    android:src="@drawable/empty"
    app:layout_constraintBottom_toTopOf="@+id/btnSubmit" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="12dp"
    android:gravity="center"
    android:text="Add Image"
    android:textColor="@color/white"
    android:textSize="16sp"
    android:textStyle="bold"
    android:translationZ="90dp"
    app:layout_constraintBottom_toBottomOf="@+id/imgItem"
    app:layout_constraintEnd_toEndOf="@+id/imgItem"
    app:layout_constraintStart_toStartOf="@+id/imgItem" />
```

Gambar 3.7 Bagian Kode Dari *Layout Upload Gambar*

Gambar 3.7 menunjukkan bagian kode dari *layout upload gambar*. Pada bagian *ImageView* terdapat sebuah id *imgItem*, yang lokasinya ditempatkan diatas sebuah tombol yang memiliki id *btnSubmit*. Sebelum *ImageView* diisi oleh gambar yang dipilih oleh user, diberikan sebuah gambar *default* yang bernama *empty*. Pada bagian *TextView*, akan menampilkan teks *Add Image* di bagian bawah sebuah gambar dari *ImageView*, teks tersebut akan berwarna putih dan teks tersebut tidak akan hilang meskipun gambar sudah berganti-ganti.

```

<Button
    android:id="@+id/btnSubmit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Submit"
    android:layout_marginStart="4dp"
    android:layout_marginEnd="4dp"
    android:layout_marginBottom="4dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

```

Gambar 3.8 Bagian Kode *Layout* Tombol Submit

Gambar 3.8 menunjukkan bagian kode layout tombol submit aplikasi. Pada bagian *Button*, diberikan id *btnSubmit*. Bagian tombol diletakkan di bagian paling bawah tampilan utama.

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:layout_marginTop="1dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toTopOf="@+id/imgItem"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/scanner_view">

```

Gambar 3.9 Bagian Kode *Layout* Bagian Dalam Aplikasi

Gambar 3.9 menunjukkan bagian kode *layout* bagian dalam aplikasi. *Layout* ini berada di dalam *layout constraint*. *LayoutConstraint* ditempatkan di bawah *scanner_view* dan diatas *imgItem*. *LinearLayout* digunakan di dalam *layout*

constraint untuk mengelompokkan bagian hasil *scan QR code* dan kondisi barang sehingga tidak mengganggu bagian *layout* lainnya.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:text="GUID ASET" />

<TextView
    android:id="@+id/textViewGUID"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:text="Id"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.Large" />
```

Gambar 3.10 Bagian Kode *Layout Hasil Scan QR Code*

Gambar 3.10 menunjukkan bagian kode *layout* hasil *scan QR Code*. Terdapat 2 jenis *TextView*, yang pertama adalah digunakan untuk memberi nama bagian hasil *scan QR Code* yaitu *GUID ASET*. Untuk *TextView* kedua yang diberi id *TextViewGUID* digunakan untuk tempat hasil *scan QR Code* ditampilkan, yang sementara berisikan *Id*.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:text="Kondisi" />

<Spinner
    android:id="@+id/spinnedKondisi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:entries="@array/condition"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"
    tools:layout_editor_absoluteX="-2dp"
    tools:layout_editor_absoluteY="298dp" />
```

Gambar 3.11 Bagian Kode *Layout Spinner Kondisi Barang*

Gambar 3.11 menunjukkan bagian kode *layout spinner* kondisi barang. Pada bagian *TextView* untuk memberi nama bagian *spinner* kondisi barang yang bernama *Kondisi*. Pada bagian *Spinner* digunakan sebagai bagian pemilih kondisi sebuah barang yang akan disimpan, id yang diberikan adah *spinnedKondisi*. Pilihan *spinner* kondisi barang adalah *Excellent*, *Good*, *Bad* dan *Very Bad* yang disimpan di *string-array* bernama *condition*.

B.3 Java

Untuk bagian Java akan menunjukkan potongan kode pemrograman yang menggunakan bahasa pemrograman Java pada aplikasi sensus barang model *QR Code scanner*, yang dibantu dengan menggunakan *screenshot*.

```
// initialize object
textViewGUID = (TextView) findViewById(R.id.textViewGUID);
spKondisi = (Spinner) findViewById(R.id.spinnedKondisi);
imgItem = (ImageView) findViewById(R.id.imgItem);
btnSubmit = (Button) findViewById(R.id.btnSubmit);
CodeScannerView scannerView = findViewById(R.id.scanner_view);
mCodeScanner = new CodeScanner(context: this, scannerView);
btnSubmit.setEnabled(false);
```

Gambar 3.12 Bagian Kode Persiapan

Gambar 3.12 menunjukkan gambar bagian kode persiapan. Bagian persiapan ini berisikan semua bagian yang akan di tampilkan ke dalam layar utama aplikasi. Bagian ini akan mengambil id yang ada di dalam file XML untuk dipakai pada bahasa pemrograman Java. Pada bagian *btnSubmit* akan membuat tombol di non-aktifkan sehingga tidak bisa langsung dipakai.

```

mCodeScanner.setDecodeCallback(new DecodeCallback() {
    @Override
    public void onDecoded(@NonNull final Result result) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                textViewGUID.setText(result.getText());
                if(toggle == 1){
                    btnSubmit.setEnabled(true);
                }else{
                    toggle = 1;
                }
            }
        });
    }
});

scannerView.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    public void onClick(View view) {
        if(checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED){
            requestPermissions(new String[]{Manifest.permission.CAMERA},MY_SCANNER_PERMISSION);
        }else {
            mCodeScanner.startPreview();
        }
    }
});

```

Gambar 3.13 Bagian Kode Utama *Scan QR Code Scanner*

Gambar 3.13 menunjukkan bagian kode utama *scan QR Code scanner*.

Ada 2 bagian, pada potongan kode ini yaitu bagian aktivasi kamera untuk *scan QR Code* dan bagian memproses hasil *scan* dan menampilkannya ke halaman utama. Pada bagian aktivasi kamera dilakukan oleh bagian *scannerView*, yang saat bagiannya di tekan oleh user terdapat 2 hasil. Jika pertama kali digunakan, aplikasi akan meminta izin untuk menggunakan kamera, jika diterima fitur *scan QR Code* baru bisa dijalankan. Jika sudah pernah diberikan izin oleh user, fitur *scan QR Code* langsung bisa digunakan. Untuk bagian proses hasil *scan QR Code* akan dilakukan pada bagian *mCodeScanner*, yang digunakan untuk men-*decode* hasil scan yang lalu akan mengubah hasil yang ada di halaman utama, menjadi hasil yang telah didapatkan pada *QR Code*. Lalu jika kondisi yang lain telah terpenuhi, tombol *submit* akan aktif.

```

private void selectImage(){
    final CharSequence[] options = { "Take Photo", "Choose from Gallery","Cancel" };
    AlertDialog.Builder builder = new AlertDialog.Builder( context MainActivity.this);
    builder.setTitle("Add Photo");
    builder.setItems(options, new DialogInterface.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.M)
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            if (options[i].equals("Take Photo"))
            {
                if(checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED){
                    requestPermissions(new String[]{Manifest.permission.CAMERA},MY_CAMERA_PERMISSION_CODE);
                }else {
                    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                    startActivityForResult(cameraIntent,CAMERA_REQUEST);
                }
            }
            else if (options[i].equals("Choose from Gallery"))
            {
                Intent photoPick = new Intent(Intent.ACTION_PICK,MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                photoPick.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, value: 1);
                startActivityForResult(photoPick,RESULT_LOAD_IMAGE);
            }
            else if (options[i].equals("Cancel")) {

```

Gambar 3.14 Bagian Kode *Upload* Gambar

Gambar 3.14 menunjukkan potongan bagian kode *upload* gambar. Bagian ini menunjukkan *function selectImage* yang akan aktif ketika ingin mengaktifkan fitur *upload* gambar. Saat dipilih akan menampilkan 3 pilihan untuk user, yaitu *Take Photo*, *Choose from Gallery* dan *Cancel*. Jika pilihan *Take Photo* dipilih, jika user pertama kali pakai, akan meminta izin untuk menggunakan kamera, jika diterima akan menjalankan tahap selanjutnya, yaitu akan menjalankan sebuah *activity* dengan mengirimkan *intent cameraIntent* yang berisikan aksi menggunakan kamera bersama *CAMERA_REQUEST* yang berisikan sebuah angka untuk mengetahui aktivitas mana yang harus dijalankan. Jika *Choose From Gallery* yang dipilih, akan menjalankan *activity* yang mengirimkan sebuah *intent* yang berisikan aksi membuka *gallery* bersama *RESULT_LOAD_IMAGE* sebagai angka aktivitasnya.

```

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                       @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if(requestCode == MY_CAMERA_PERMISSION_CODE){
        if(grantResults[0] == PackageManager.PERMISSION_GRANTED){
            Toast.makeText(context, this, "Camera Permission Granted", Toast.LENGTH_SHORT).show();
            Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(cameraIntent,CAMERA_REQUEST);
        }
    }else if (requestCode == MY_SCANNER_PERMISSION){
        if(grantResults[0] == PackageManager.PERMISSION_GRANTED){
            Toast.makeText(context, this, "Camera Permission Granted", Toast.LENGTH_SHORT).show();
            mCodeScanner.startPreview();
        }
    }
}
}

```

Gambar 3.15 Bagian Kode *Permission* Kamera

Gambar 3.15 menunjukkan bagian kode *permission* atau izin penggunaan kamera. Potongan kode ini akan dijalankan ketika user ingin menggunakan fitur kamera pertama kalinya, yang akan mengirimkan permintaan *permission*. Jika diterima, akan memunculkan sebuah *toast*, bahwa aplikasi telah mendapat izin.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (resultCode == RESULT_OK) {
        if (requestCode == CAMERA_REQUEST && resultCode == Activity.RESULT_OK) {
            Bitmap photo = (Bitmap) data.getExtras().get("data");
            Bitmap resized = Bitmap.createScaledBitmap(photo, dstWidth: 800, dstHeight: 800, filter: true);
            imgItem.setImageBitmap(resized);
            if(toggle == 1){
                btnSubmit.setEnabled(true);
            }else{
                toggle = 1;
            }
        } else if (requestCode == RESULT_LOAD_IMAGE) {
            imageUrl = data.getData();
            imgItem.setImageURI(imageUri);
            if(toggle == 1){
                btnSubmit.setEnabled(true);
            }else{
                toggle = 1;
            }
        }
    }
}

```

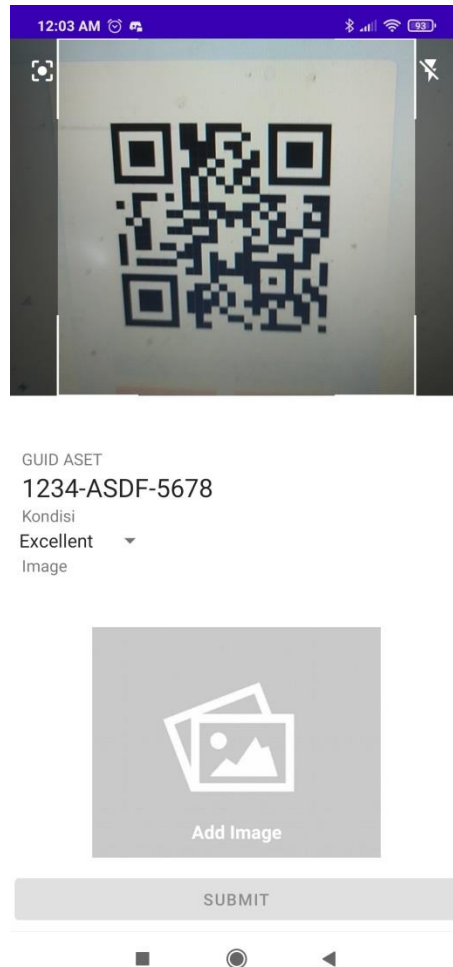
Gambar 3.16 Bagian Kode *Activity Upload Gambar*

Gambar 3.16 menunjukkan gambar bagian kode *activity upload* gambar. Saat sebuah permintaan dari *fuction selectImage* dijalankan, dan hasil *requestCode* diterima, aktivitas yang berdasarkan nilai *requestCode* tersebut akan dijalankan. Jika *requestCode* yang berisikan nilai *CAMERA_REQUEST* dijalankan, aplikasi akan menjalankan fitur kamera yang akan mengambil sebuah gambar, gambar tersebut jika telah dikonfirmasi akan mengganti gambar yang sebelumnya ada di halaman utama aplikasi. Jika *requestCode* yang berisikan nilai *RESULT_LOAD_IMAGE* dijalankan, akan membuat *gallery* yang dapat memilih 1 gambar untuk dipilih, setelah dipilih aplikasi akan mengambil rute lokasi gambar, dan mengambilnya untuk mengubah gambar yang ada di halaman utama aplikasi.

B.4 Pengujian

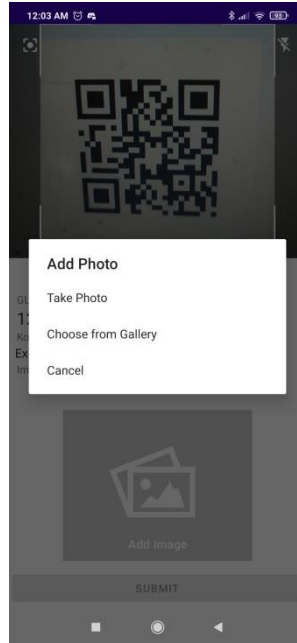
Pada bagian ini, akan menunjukkan contoh *flow* penggunaan aplikasi Sensus Barang modul *QR Code Scanner*, yang akan menggunakan *screenshot* sebagai pembantu.

Saat aplikasi dimulai, halaman seperti pada Gambar 3.4 akan muncul, semua data berisikan kosong, dan tombol submit non-aktif.



Gambar 3.17 Hasil *Scan QR Code*

User lalu menekan daerah kamera fitur *QR Code*, fitur tersebut akan selalu menyala sampai *user* menemukan sebuah *QR Code*. Ketika *QR Code* telah ditemukan, kamera akan berhenti dan akan memproses data yang ada di dalam *QR Code*. Data yang telah diambil akan ditampilkan pada daerah *GUID ASET*, seperti pada Gambar 3.17.



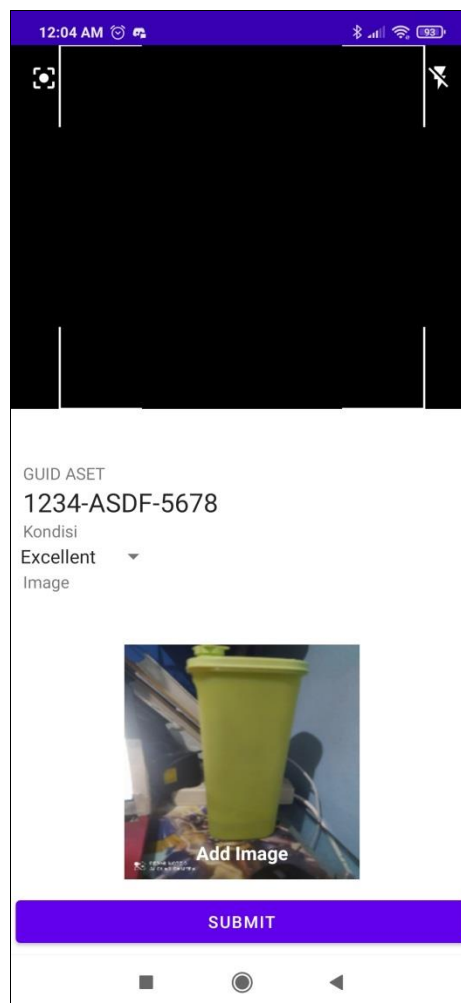
Gambar 3.18 Gambar *Popup* Pemilihan Jenis Upload

Lalu user harus mengupload sebuah gambar barang tersebut dengan menekan area *add Image*, yang nantinya akan memunculkan menu seperti di Gambar 3.18.



Gambar 3.19 Konfirmasi Pengambilan Gambar Menggunakan Kamera

Ketika pilihan *Take Photo* ditekan, *user* akan dibawa ke fitur kamera. Ketika *user* telah mengambil gambar menggunakan kamera tersebut, *user* akan dibawa ke bagian konfirmasi pengambilan gambar seperti Gambar 3.19. Jika diterima gambar akan dibawa kembali ke halaman utama dan gambar akan diganti seperti apa yang telah kamera ambil seperti di Gambar 3.20.

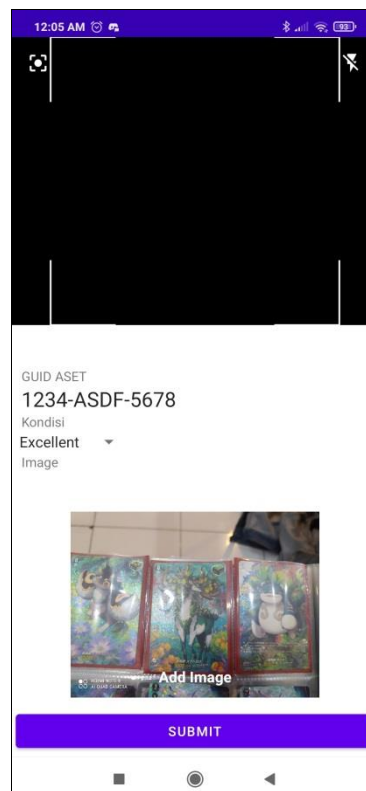


Gambar 3.20 Hasil Perubahan Gambar

Jika user memilih menu *Choose From Gallery*, user akan dibawa ke *gallery* untuk diminta untuk memilih gambar seperti Gambar 3.21.



Gambar 3.21 Menu Gallery



Gambar 3.21 Hasil Perubahan Gambar

Lalu ketika gambar telah berubah seperti Gambar 3.21, *user* dapat menekan tombol *submit* untuk menyimpan data yang telah diberikan.

3.3.3. Kendala yang Ditemukan

Dalam pembuatan aplikasi sensus barang modul *QR Code scanner* pada Badan Pengelolaan Aset Daerah DKI Jakarta terdapat beberapa kendala yang dialami, yaitu sebagai berikut:

1. Kurangnya pengalaman dalam menggunakan *IDE* Android Studio dan Library yang telah diberikan, sehingga memerlukan waktu yang lama untuk menyelesaikan permintaan.
2. Komputer yang penulis gunakan merupakan model yang cukup lama, sehingga beberapa proses seperti *build* harus menunggu waktu yang cukup lama.
3. Beberapa fitur yang diminta belum selesai dikembangkan dikarenakan limitasi waktu dan persiapan yang belum tersedia.

3.3.4. Solusi Atas Kendala yang Ditemukan

Solusi yang dilakukan untuk mengatasi kendala dalam pembuatan aplikasi sensus barang modul *QR Code scanner* adalah:

1. Melakukan pembelajaran mandiri melalui dokumentasi yang ada di *Stack Overflow*, dan Medium jika menemukan masalah. Selain itu menggunakan metode *Trial and Error*.
2. Berdiskusi dengan supervisi tentang perubahan yang bisa digantikan, agar cocok dengan permintaan akhir.