

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Aplikasi

Rancangan aplikasi ini dilakukan dengan menggunakan Use Case Diagram yang meliputi activity diagram, sequence diagram. Aplikasi pemesanan banner di CV. Andalan Kita berbasis android dengan nama aplikasi Andalan Kita menggunakan metode *Rapid Application Development* (RAD) dengan tahapan requirement planning, user design, build system, implementation. Pelaksanaan penelitian ini secara lengkap dapat dilihat sebagai berikut :

4.1.1 Requirement Planning

Pada tahap ini dilakukan analisis mengenai kebutuhan dalam pembuatan aplikasi yang meliputi analisis kebutuhan data dan informasi, analisis alat yang digunakan, analisis fungsionalitas dan non fungsionalitas sistem dalam membuat aplikasi.

4.1.1.1 Analisis Kebutuhan Data dan Informasi

Sistem ini membutuhkan data dan informasi berupa data dan informasi detail produk yang ada di CV. Andalan Kita. Untuk data yang diperlukan seperti pilihan produk, harga produk, jenis finishing, dan foto produk di CV. Andalan Kita.

4.1.1.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan analisis beberapa fungsi yang diperlukan dalam sistem aplikasi berdasarkan angket responden terhadap beberapa pengguna yang telah dilakukan, fungsi yang diperlukan dalam sistem aplikasi ini antara lain :

1. Pengguna dapat menjalankan aplikasi ini dimana dan kapan saja dengan akses jaringan internet yang stabil.
2. Pengguna dapat melakukan pemesanan produk tanpa harus datang ke lokasi percetakan.
3. Sistem aplikasi ini memiliki fitur status pemesanan, memudahkan pengguna mengetahui informasi produk yang telah dipesan.

4.1.1.3 Analisis Kebutuhan Non Fungsionalitas

Kebutuhan non fungsionalitas merupakan kebutuhan yang tidak langsung berhubungan dengan spesifik yang disediakan oleh sistem. Kebutuhan ini berhubungan dengan properti sistem yang muncul belakangan, seperti keandalan, waktu tanggap dan penempatan pada media penyimpanan. Kebutuhan non fungsionalitas aplikasi ini antara lain :

1. Aplikasi ini dapat berjalan di sistem operasi android minimal versi 4.4 (*Kitkat*).
2. Aplikasi ini akan berjalan lebih optimal apabila dijalankan pada *device* dengan ukuran layar 6,3 inchi 1080 x 2340 pixel.

4.1.1.4 Analisis Kebutuhan Alat

Peralatan yang digunakan untuk membuat aplikasi Andalan Kita berbasis android ini antara lain :

1) Perangkat keras (*Hardware*)

Adapun perangkat keras (*Hardware*) yang digunakan dalam membantu pembuatan aplikasi ini berupa laptop dan smartphone dengan spesifikasi sebagai berikut:

a. Laptop

- Operating system : Windows 7 64bit
- Processor : Intel(R) Core(TM) i5-4200U @1.60GHz
- Memory : 8.00 GB RAM

- Harddisk : 500 GB
- b. Smartphone
 - Sistem operasi : Android Pie
 - Jaringan : GSM/ CDMA/ LTE
 - Processor : Octa-core 2.2 GHz
 - Memory : 3,00 GB RAM
 - Ukuran Layar : 6,3 inchi 1080x2340 Pixel

2) Perangkat lunak (*Software*)

Perangkat lunak yang digunakan laptop adalah :

- Operating system : Windows 7 64bit
- Android Studio
- Corel Draw X7

3) Data

Adapun data-data yang dibutuhkan dalam pembangunan aplikasi Andalan Kita antara lain :

- a. Data harga banner
- b. Data jenis *finishing*
- c. Foto banner

4.1.2 User Desain

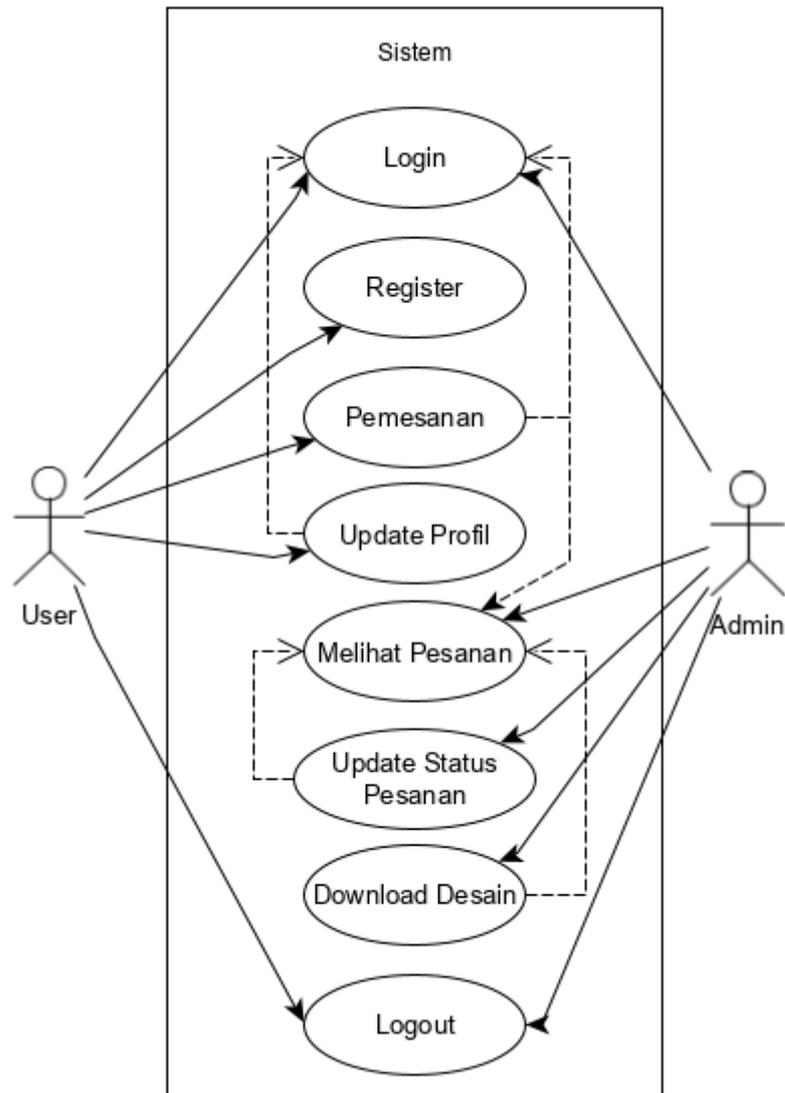
4.1.2.1 Deskripsi Sistem

Untuk dapat menjelaskan aplikasi Andalan Kita ini secara mudah dan sederhana, maka pada tahap ini penulis membuat perancangan system dengan menggunakan diagram UML (*Unified Modelling Language*) yang di gambarkan melalui perancangan diagram seperti use case diagram, activity diagram, dan sequence diagram.

4.1.2.2 Use Case Diagram

Use case diagram mendefinisikan fitur fitur yang terdapat dalam Aplikasi Andalan Kita. Use case diagram menunjukkan adanya interaksi antara user dan sistem, dimana

user adalah pengguna dan system adalah aplikasi Andalan Kita. Use case diagram dapat dilihat pada gambar berikut ini :



Gambar 4.1. Use Case Diagram

Deskripsi use case diagram dari gambar diatas di jelaskan lebih detail pada table berikut :

Tabel 4.1. Deskripsi Use Case Diagram User

User (Pengguna)		
No.	Use Case	Deskripsi
1.	Register	Pengguna melakukan pendaftar akun sebelum membuka aplikasi
2.	Login	Pengguna melakukan login aplikasi jika sudah pernah melakukan pendaftaran akun sebelumnya
3.	Pemesanan	Pengguna memilih dan mengisi form produk yang dipesan
4.	Update Profil	Pengguna melengkapi atau mengganti data diri
5.	Logout	Pengguna melakukan keluar dari akun dan aplikasi

Tabel 4.2. Deskripsi Use Case Diagram Admin

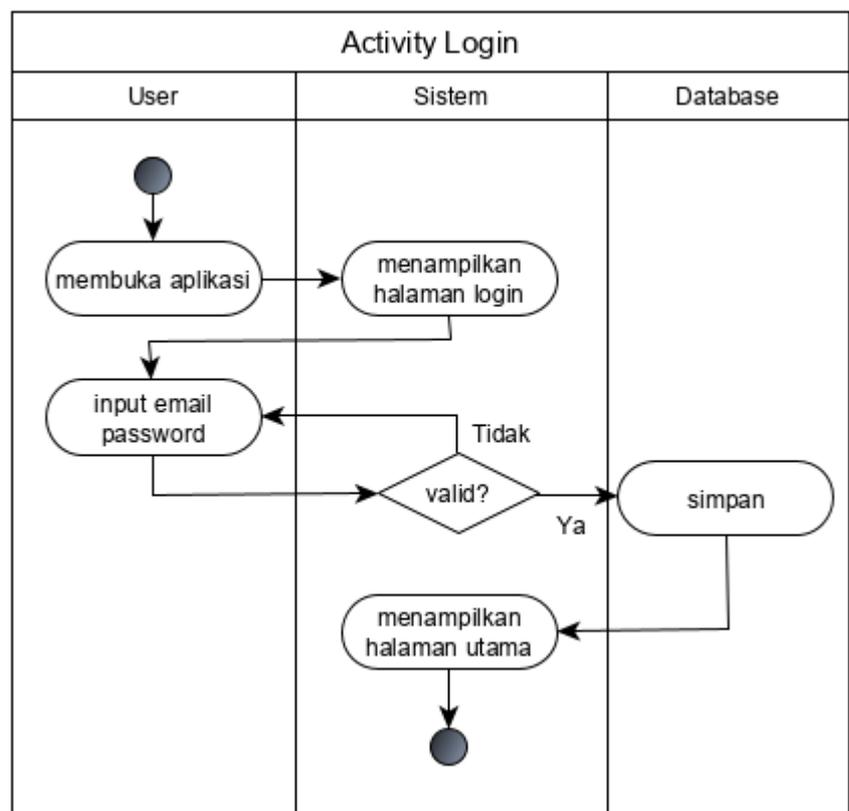
Admin		
No.	Use Case	Deskripsi
1.	Login	Admin melakukan login sebelum membuka aplikasi
2.	Melihat pesanan	Admin melihat daftar pesanan pelanggan
3.	Uupdate status pesanan	Admin memberi konfirmasi status pesanan pelanggan
4.	Download desain	Admin mengunduh desain gambar dari pelanggan
5.	Logout	Admin melakukan keluar dari akun dan aplikasi

4.1.2.3 Activity Diagram

Activity diagram sebagai gambaran tentang bagaimana kinerja sistem aplikasi saat pengguna menjalankannya. *Activity diagram* lebih berfokus terhadap aktivitas apa saja yang dilakukan oleh *system* bukan apa yang dilakukan oleh *user*. Berikut adalah activity diagram dari Aplikasi Andalan Kita.

a. Aplikasi User

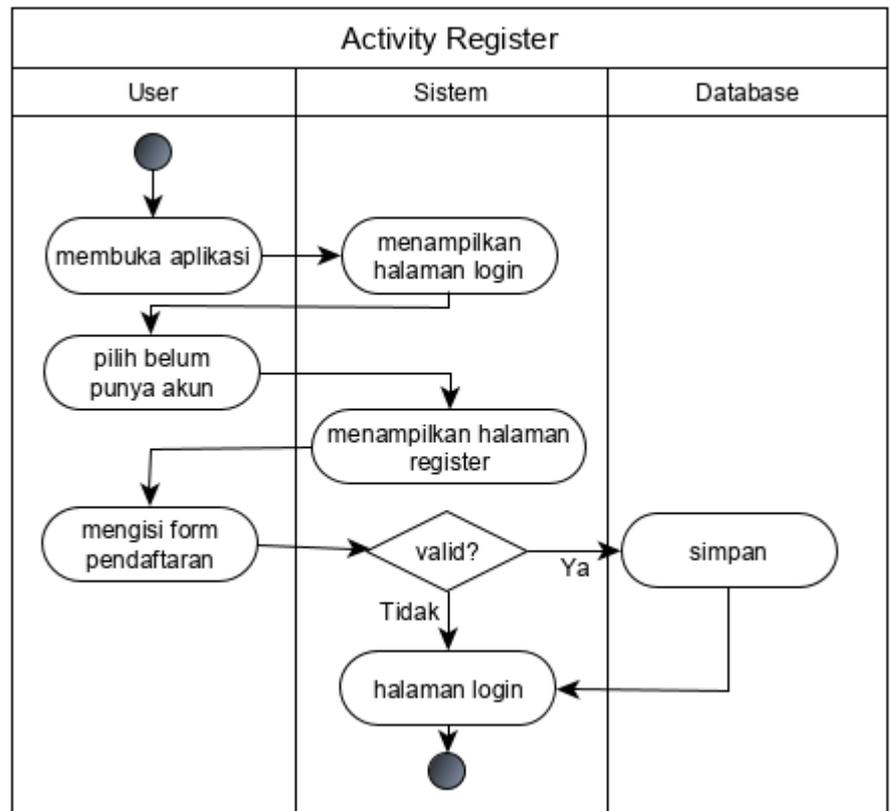
1. Activity Diagram Login User



Gambar 4.2. Activity Diagram Login

Pada *activity login user*, *user* membuka aplikasi dan sistem menampilkan halaman login dan kemudian input data login dengan *username* dan *password* akan memvalidasi data *user* jika data sudah benar maka akan disimpan dan selanjutnya menampilkan halaman utama.

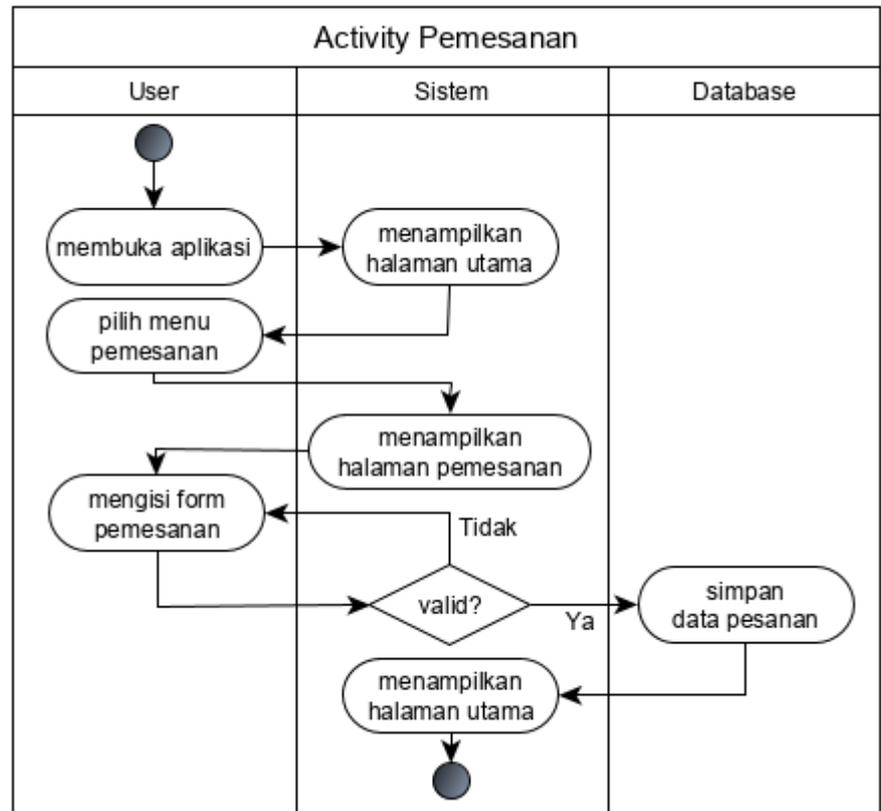
2. Activity Diagram Register User



Gambar 4.3. Activity Diagram Register User

Pada *activity register user* ini pertama membuka aplikasi lalu sistem akan menampilkan halaman login setelah itu *user* memilih daftar, lalu sistem akan menampilkan halaman register dengan form registrasi, setelah itu melakukan input data user, sistem akan memvalidasi jika iya maka data *user* akan disimpan dan langsung ke menu halaman login apabila tidak maka data tidak akan disimpan dan menampilkan langsung ke halaman login.

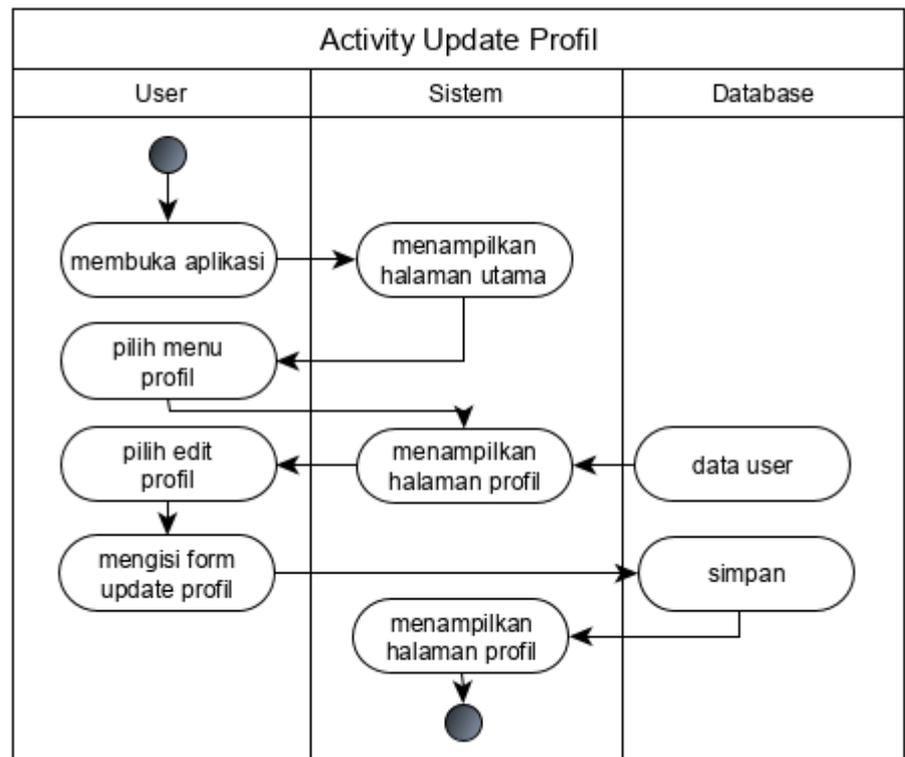
3. Activity Diagram Pemesanan



Gambar 4.4. Activity Diagram Daftar Pesanan

Pada *activity* pemesanan, pertama user melakukan login, lalu system akan menampilkan menu halaman utama. Kemudian user memilih menu pemesanan dan melakukan input data seperti nama pemesanan, nama produk dan ukuran. Sistem akan memvalidasi jika iya maka data *user* akan disimpan.

4. Activity Update Profil

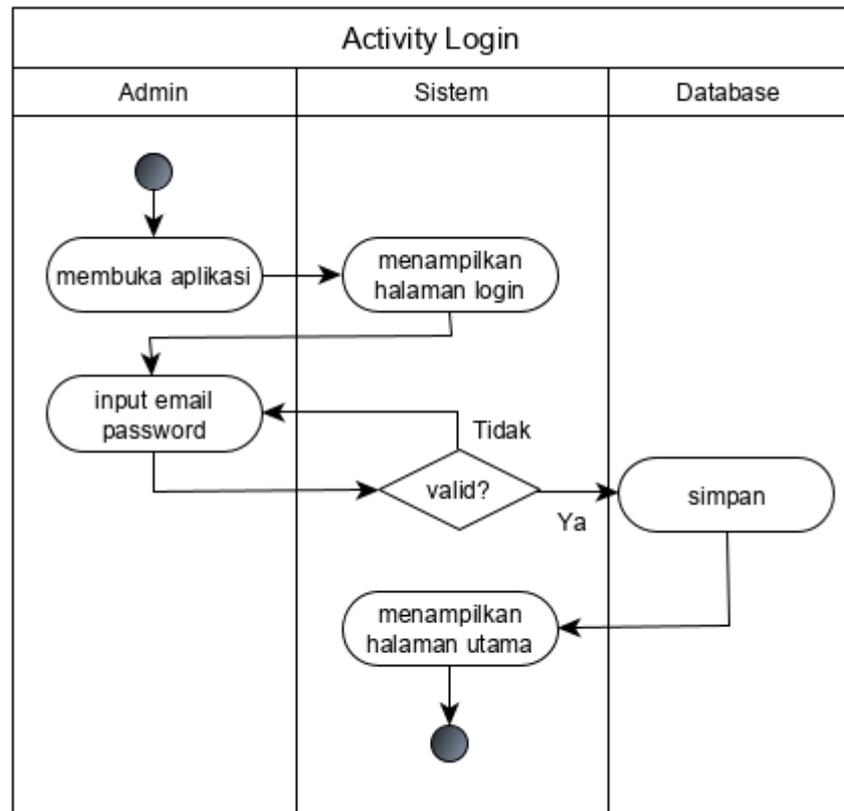


Gambar 4.5. Activity Diagram Update Profil

Pada activity update profil, pertama user melakukan login, lalu sistem akan menampilkan halaman utama. Kemudian user klik menu profil dan input data apa saja yang perlu diubah untuk pembaruan dari data sebelumnya.

b. Aplikasi Admin

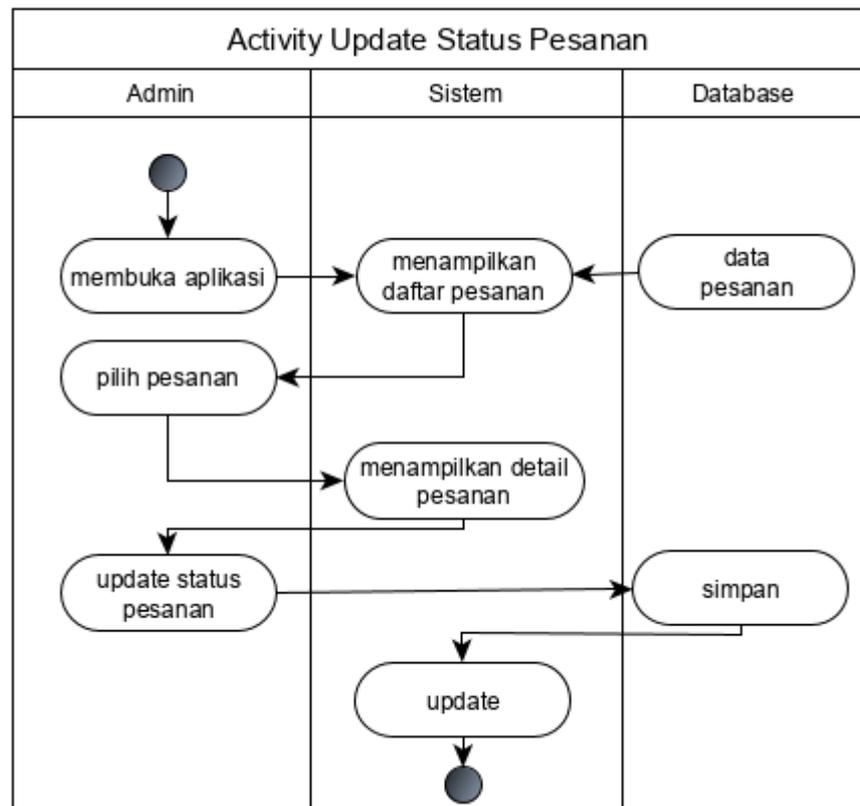
1. Activity Diagram Login Admin



Gambar 4.6. Activity Diagram Login Admin

Pada activity login admin, pertama admin membuka aplikasi dan memasukan email dan password untuk bisa masuk ke halaman utama aplikasi.

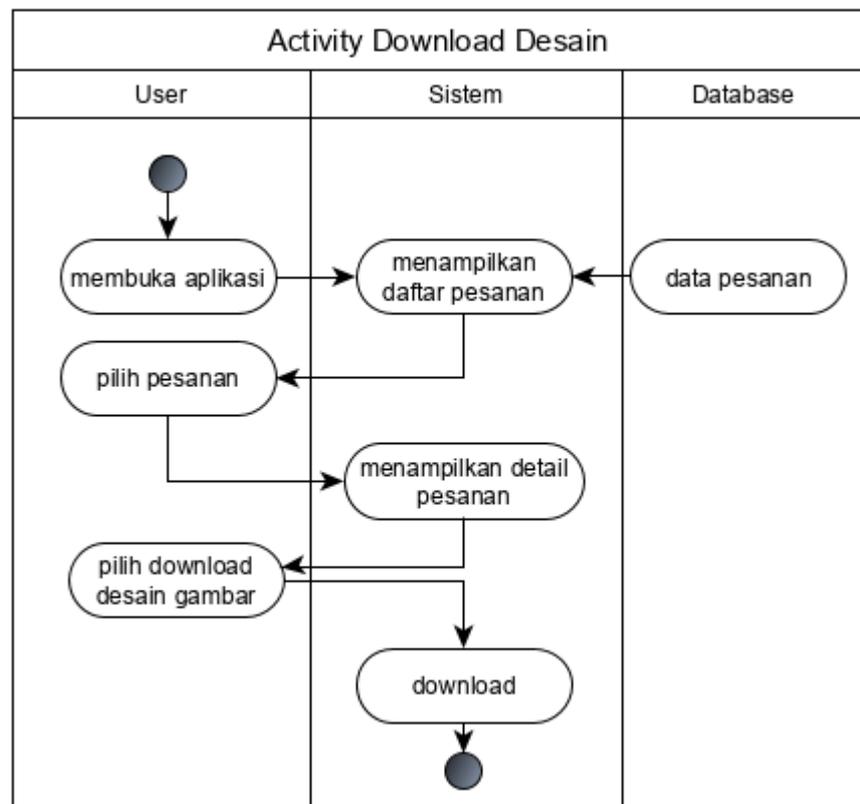
2. Activity Diagram Update Status Pesanan



Gambar 4.7. Activity Diagram Update Status Pesanan

Pada activity update status pesanan, pertama admin melakukan login, setelah itu system akan menampilkan halaman daftar pesanan. Kemudian admin memilih pesanan, selanjutnya admin melakukan konfirmasi informasi pesanan.

3. Activity Diagram Download Desain



Gambar 4.8. Activity Diagram Download Desain

Pada activity *download* desain, pertama admin melakukan login, setelah itu system akan menampilkan halaman daftar pesanan. Kemudian admin memilih pesanan, selanjutnya admin mengunduh gambar desain user.

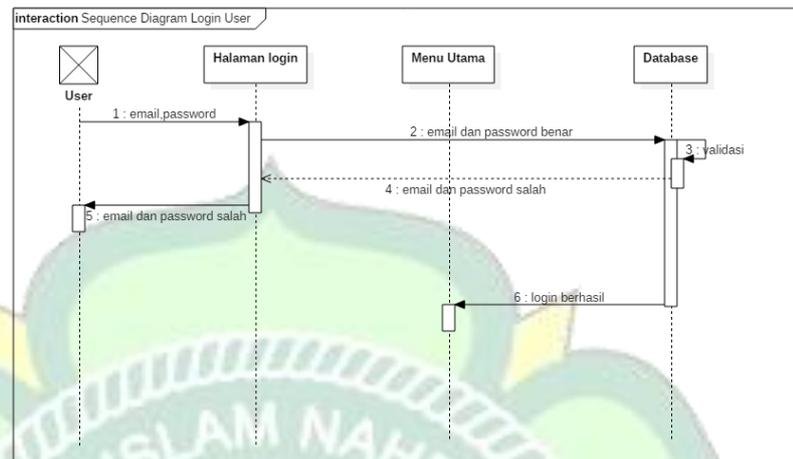
4.1.2.4 Sequence Diagram

Sequence diagram merupakan penggambaran rangkaian langkah langkah yang dilakukan sebagai respon dari sebuah elemen untuk menghasilkan keluaran tertentu. Sequence diagram disusun berdasarkan urutan waktu. Berikut adalah sequence diagram dari Aplikasi Andalan Kita.

a. Aplikasi User

1. Sequence Diagram Login User

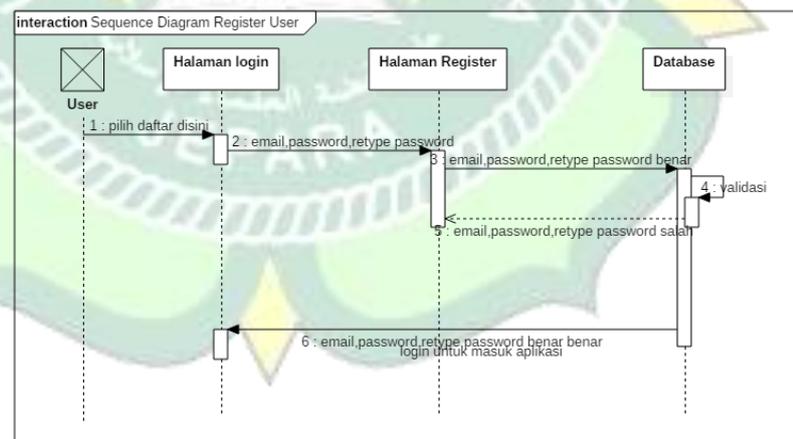
Pada diagram login merupakan proses login dan dilakukan oleh user dengan memasukkan email dan password.



Gambar 4.9. Sequence Diagram Login User

2. Sequence Diagram Register User

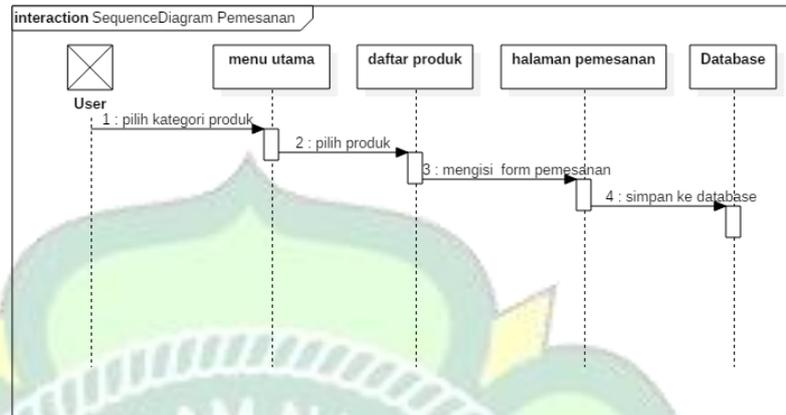
Diagram ini merupakan proses pendaftaran user menggunakan email dan password, jika berhasil mendaftar masuk ke tahap login.



Gambar 4.10. Sequence Diagram Register User

3. Sequence Diagram Pemesanan

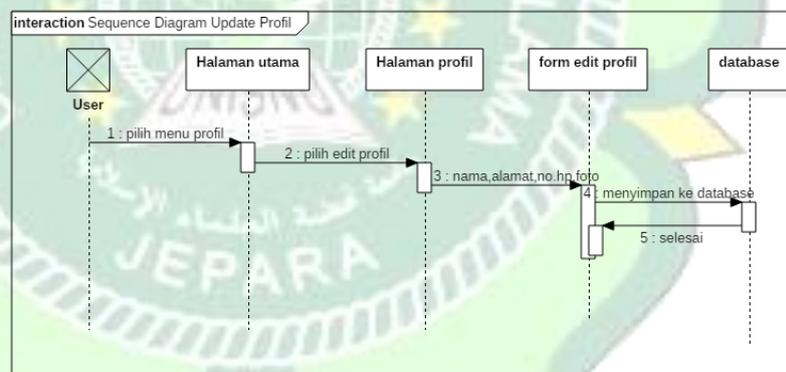
Diagram ini merupakan proses user memilih pesanan dan mengisi form pemesanan.



Gambar 4.11. Sequence Diagram Pemesanan

4. Sequence Diagram Update Profil

Diagram ini merupakan proses user melakukan perubahan atau melengkapi profil.

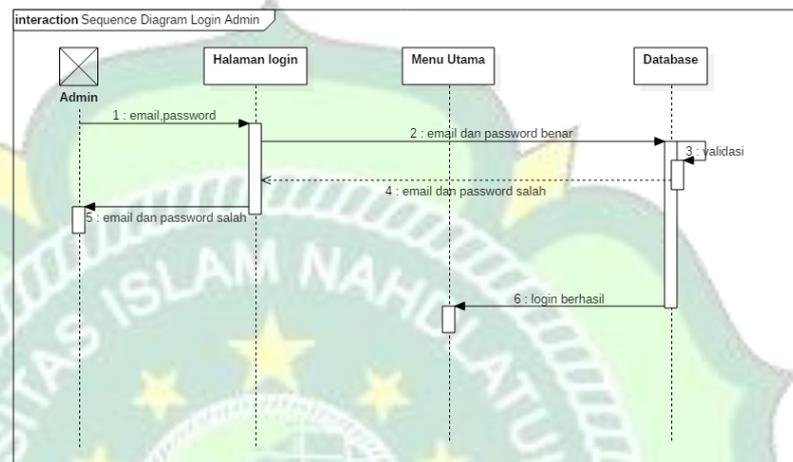


Gambar 4.12. Sequence Diagram Update Profil

b. Aplikasi Admin

1. Sequence Diagram Login Admin

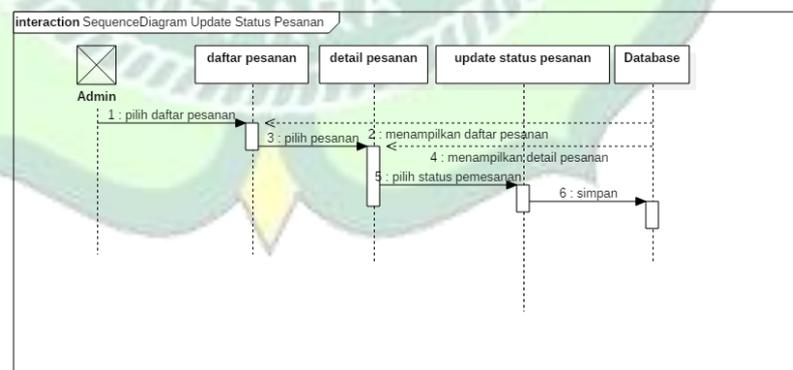
Pada diagram ini merupakan proses login admin dengan memasukan email dan password sebelum masuk ke halaman utama.



Gambar 4.13. Sequence Diagram Login Admin

2. Sequence Diagram Update Status Pesanan

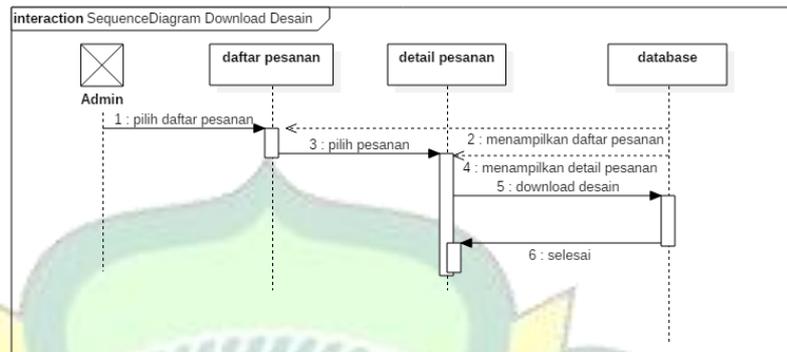
Pada diagram ini merupakan proses melihat dan konfirmasi informasi pesanan dari user yang dilakukan oleh admin.



Gambar 4.14. Sequence Diagram Update Pesanan

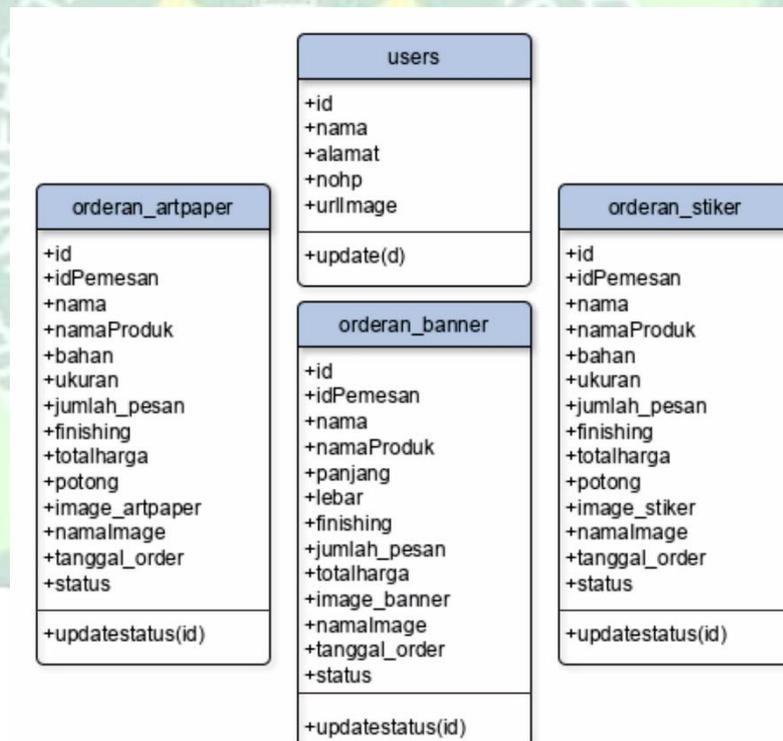
3. Sequence Diagram Download Desain

Pada diagram ini merupakan proses mengunduh desain gambar dari user yang dilakukan oleh admin.



Gambar 4.15. Sequence Diagram Download Gambar

4.1.2.5 Class Diagram



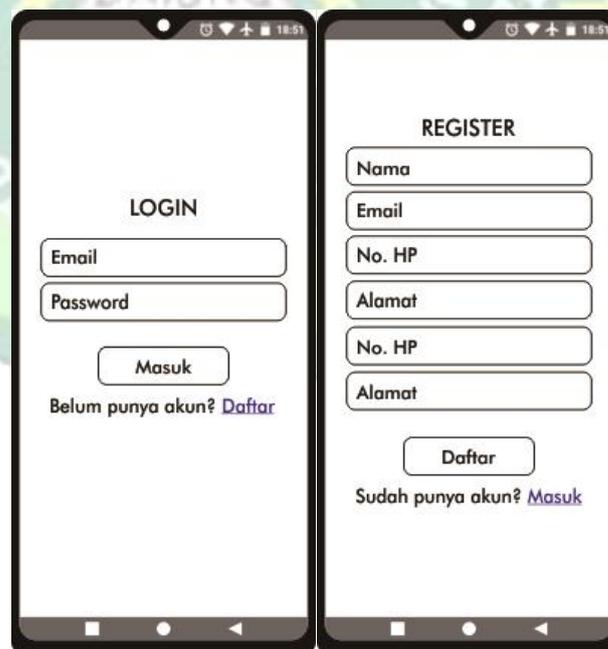
Gambar 4.16. Class Diagram



Gambar 4.18. Rancangan Splash Screen

2. Rancangan *Login/SignUp User*

Halaman *Login/SignUp* merupakan halaman masuk atau mendaftar aplikasi menggunakan *email* dan *password*. Perancangan tampilan dari halaman ini dapat dilihat pada gambar berikut :



Gambar 4.19. Rancangan Login/Signup User

Penjelasan gambar :

Pertama user diminta untuk *login* menggunakan *email* dan *password*, jika belum mempunyai akun pengguna mendaftarkan *email*, nama, dan *password* di menu daftar atau *register*.

3. Rancangan Halaman Beranda

Halaman beranda ini adalah halaman yang muncul setelah halaman login. Halaman Beranda ini terdapat pilihan pesan banner, pesan artpaper dan pesan stiker. Berikut rancangan beranda *user* :



Gambar 4.20. Rancangan Beranda User

4. Rancangan Menu Pemesanan

Pada menu ini pengguna mengisi form pemesanan termasuk mengunggah desain gambar yang akan dicetak. Rancangan menu pemesanan dapat di lihat pada gambar berikut :

Nama Pemesan
 Nama Produk
 Ukuran x
 Jumlah Cetak
 Finishing

Pilih Gambar

PESAN

Home Pesanan Profil

Gambar 4.21. Rancangan Menu Pemesanan

5. Rancangan Menu Pesanan

Pada menu ini user bisa mengetahui riwayat pemesanan dari produk yang baru dipesan, pernah dipesan sebelumnya, dan informasi pemesanan. User harus menunggu konfirmasi dari admin terlebih dahulu mengetahui informasi pesannya. Rancangan menu ini dapat dilihat pada gambar berikut :

PESANAN

Nama Pemesan
 No. HP
 Tanggal Pesan
 Nama Produk
 Ukuran
 Finishing
 Jumlah Cetak
 Total Harga

Status Pemesanan

Sedang Proses Pencetakan

Gambar 4.22. Rancangan Menu Pesanan

6. Rancangan Menu Profil

Pada menu ini terdapat informasi data diri dan memperbarui data diri user. Rancangan menu ini dapat dilihat pada gambar berikut:

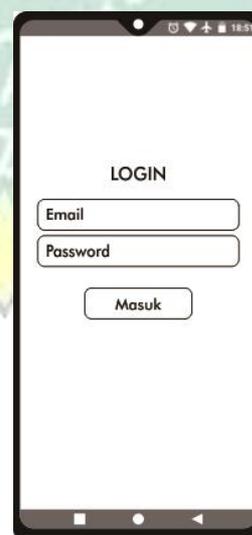


Gambar 4.23. Rancangan Menu Profil

b. Rancangan aplikasi admin

1. Rancangan halaman *login* admin

Pada halaman ini admin mengisi email dan password admin. Rancangan menu ini dapat dilihat pada gambar berikut:



Gambar 4.24. Rancangan Halaman Login Admin

2. Rancangan Halaman Melihat Pesanan

Pada halaman ini admin melakukan memilih daftar pesanan yang masuk. Rancangan menu ini dapat dilihat pada gambar berikut :



Gambar 4.25. Rancangan Halaman Melihat Pesanan

3. Rancangan Halaman Update Pesanan

Pada halaman ini admin melihat deskripsi pesanan, konfirmasi status atau informasi pemesanan dan mengunduh desain gambar dari user. Rancangan menu ini dapat dilihat pada gambar berikut :

Gambar 4.26. Rancangan Halaman Konfirmasi Pesanan

4.1.3 Build System (Membangun Sistem)

Pada tahap ini penulis pembangun sistem aplikasi sesuai dengan perancangan sistem yang telah penulis lakukan sebelumnya. Penulis membangun sistem sesuai dengan perancangan yang sudah penulis buat kedalam bentuk kode pemrograman.

1. Analisis sistem *login/signup user*

Login/signup merupakan halaman untuk akses masuk atau mendaftar akun ke aplikasi. Berikut adalah potongan kode yang digunakan untuk login/signup.

```
private fun loadLogin(email:String,password:String) {
    mProgress.setMessage("Loading . .")
    mProgress.setCanceledOnTouchOutside(false)
    mProgress.show()
    mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener { task :Task<AuthResult!> ->
        when {
            task.isSuccessful -> {
                mProgress.dismiss()
                startActivity(Intent( packageContext: this, MainActivity::class.java))
                finish()
            }
            mAuth.currentUser?.isEmailVerified != true -> {
                mProgress.dismiss()
                Toast.makeText( context: this, text: "Please Verify Your Email", Toast.LENGTH_LONG).show()
            }
            else -> {
                Toast.makeText( context: this, task.exception?.message.toString(), Toast.LENGTH_LONG).show()
                mProgress.dismiss()
            }
        }
    }
}
```

```

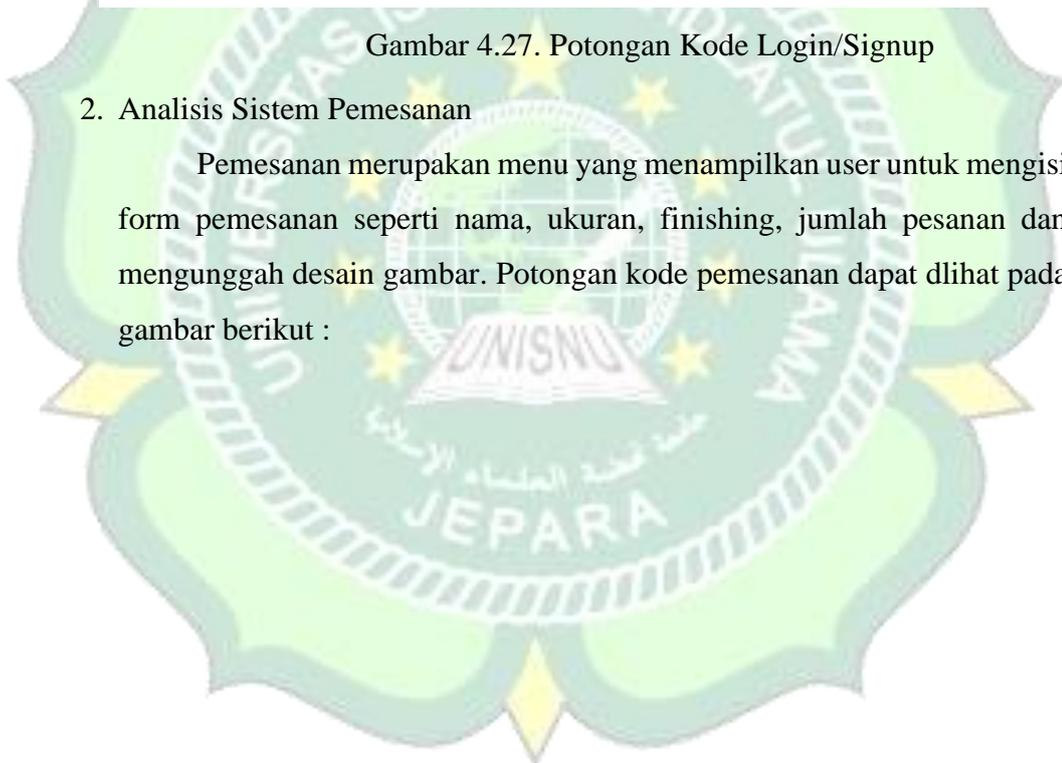
 mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener {task :Task<AuthResult> ->
    if (task.isSuccessful) {
        mAuth.currentUser?.sendEmailVerification()?.addOnCompleteListener {task :Task<Void> ->
            if (task.isSuccessful) {
                val data =
                    ModelUser(FirebaseAuth.getInstance().currentUser?.uid.toString(), nama, alamat, nohp, urlImage: "")
                FirebaseFirestore.getInstance().document(
                    documentPath: "users/${FirebaseAuth.getInstance().currentUser?.uid}
                    ): throw NullPointerException("UID is null}")
                ).set(data).addOnSuccessListener { it:Void!
                    mAuth.signOut()
                    emailRegister.text?.clear()
                    passwordRegister.text?.clear()
                    namaReg.text?.clear()
                    alamatRegister.text?.clear()
                    NohpRegister.text?.clear()
                    confirmPasswordRegister.text?.clear()
                    Toast.makeText(context: this, text: "Berhasil Registrasi", Toast.LENGTH_LONG).show()
                    mProgress.dismiss()
                }
            } else {
                Toast.makeText(context: this, text: "Register Failed", Toast.LENGTH_LONG).show()
                mProgress.dismiss()
            }
        }
    } else {
        Toast.makeText(context: this, task.exception?.message.toString(), Toast.LENGTH_LONG).show()
        mProgress.dismiss()
    }
}
}

```

Gambar 4.27. Potongan Kode Login/Signup

2. Analisis Sistem Pemesanan

Pemesanan merupakan menu yang menampilkan user untuk mengisi form pemesanan seperti nama, ukuran, finishing, jumlah pesanan dan mengunggah desain gambar. Potongan kode pemesanan dapat dilihat pada gambar berikut :



```

alert.setPositiveButton( text: "Yes") { _: DialogInterface?, _ :Int ->

    if (::selectedImageBytes.isInitialized){
        mProgress.setMessage("Loading . .")
        mProgress.setCanceledOnTouchOutside(false)
        mProgress.show()
        refStorage.putFile(selectImagePath!!).addOnSuccessListener { it: UploadTask.TaskSnapshot!
            refStorage.downloadUrl.addOnSuccessListener { it: Uri!
                val order = ModelOrderBanner(id, idPemesan, nama, namaproduk, panjangbanner, lebarbaner, finishing, jumlahpesan,
                    hargatotal.toString(), it.toString(), filename2, currentDate, status: 0)
                FirebaseFirestore.getInstance().collection( collectionPath: "orderan_banner").document(id).set(order)
                .addOnCompleteListener { task :Task<Void> ->
                    if (task.isSuccessful){...}else{
                        mProgress.dismiss()
                        Toast.makeText( context: this, text: "Gagal", Toast.LENGTH_LONG).show()
                    }
                }
            }
        }
    }
}
else{
    Snackbar.make(buatpesanbaner, text: "Silahkan upload desain anda", Snackbar.LENGTH_LONG).show()
}
}

```

```

alert.setPositiveButton( text: "Yes") { _: DialogInterface?, _ :Int ->
    if (::selectedImageBytes.isInitialized){
        mProgress.setMessage("Loading . .")
        mProgress.setCanceledOnTouchOutside(false)
        mProgress.show()
        refStorage.putFile(selectImagePath!!).addOnSuccessListener { it: UploadTask.TaskSnapshot!
            refStorage.downloadUrl.addOnSuccessListener { it: Uri!
                if (checkBoxPaper.isChecked) {
                    val order = ModelOrderArtpaper(id, idPemesan, nama, namaproduk,
                        bahan, ukuran, jumlahpesan, finishing, hargatotal.toString(),
                        potong: "Dipotong", it.toString(), filename2, currentDate, status: 0)
                    FirebaseFirestore.getInstance().collection( collectionPath: "orderan_artpaper").document(id).set(order)
                    .addOnCompleteListener {...}
                }else{
                    val order = ModelOrderArtpaper(id, idPemesan, nama, namaproduk,
                        bahan, ukuran, jumlahpesan, finishing, hargatotal.toString(),
                        potong: "Tidak Dipotong", it.toString(), filename2, currentDate, status: 0)
                    FirebaseFirestore.getInstance().collection( collectionPath: "orderan_artpaper").document(id).set(order)
                    .addOnCompleteListener { task :Task<Void> ->
                        if (task.isSuccessful) {...} else {
                            mProgress.dismiss()
                            Toast.makeText( context: this, text: "Gagal", Toast.LENGTH_LONG).show()
                        }
                    }
                }
            }
        }
    }
}
}

```


4. Analisis Sistem Update Status Pesanan

Menu update status pesanan pada aplikasi ini berfungsi untuk memberikan informasi pemesanan. Potongan kode update status pesanan dapat dilihat pada gambar berikut :

```

24 class MapsFragment : Fragment(), OnMapReadyCallback, GoogleMap.OnMarkerClickListener {
25
26     private lateinit var mMap: GoogleMap
27     private var latitude: Double = 0.0
28     private var longitude: Double = 0.0
29     private var judul: String = ""
30
31     companion object {
32         const val LOCATION_PERMISSION_REQUEST_CODE = 1
33         var mapFragment : SupportMapFragment? = null
34     }
35
36     override fun onCreateView(
37         inflater: LayoutInflater, container: ViewGroup?,
38         savedInstanceState: Bundle?
39     ): View? {
40         // Inflate the layout for this fragment
41         latitude = activity?.intent?.getStringExtra(AppConstants.LATITUDE)?.toDouble()!!
42         longitude = activity?.intent?.getStringExtra(AppConstants.LONGITUDE)?.toDouble()!!
43         judul = activity?.intent?.getStringExtra( name: "title").toString()
44         val idHome = activity?.intent?.getStringExtra(AppConstants.ID_HOMESTAY)
45         val view = inflater.inflate(R.layout.fragment_maps, container, attachToRoot = false)
46         mapFragment = childFragmentManager.findFragmentById(R.id.map) as SupportMapFragment
47         mapFragment?.getMapAsync(this)
48
49         return view
50     }
51
52     @SuppressWarnings("MissingPermission")
53     override fun onMapReady(googleMap: GoogleMap?) {
54         mMap = googleMap!!
55         mMap.isMyLocationEnabled = true
56         mMap.uiSettings.isZoomControlsEnabled = true
57         mMap.setOnMarkerClickListener(this)
58         mMap.addMarker(MarkerOptions().position(LatLng(latitude, longitude)).title(judul))
59         mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(LatLng(latitude, longitude), 12.0f))
60         setUpMap()
61     }
62
63     private fun setUpMap() {
64         if (ActivityCompat.checkSelfPermission(this@MapsFragment.context!!,
65             android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
66             ActivityCompat.requestPermissions(this@MapsFragment.activity!!,
67                 arrayOf(android.Manifest.permission.ACCESS_FINE_LOCATION), LOCATION_PERMISSION_REQUEST_CODE)
68             return
69         }
70     }
71
72     override fun onMarkerClick(p0: Marker?) = false
73 }

```

Gambar 4.30. Potongan Kode Update Status Pesanan

4.1.4 Implementation (implementasi)

Implementasi sistem merupakan tahap penerapan sistem yang akan dilakukan sesuai dengan perancangan yang telah penulis buat sebelumnya dengan menggunakan bahasa pemrograman Kotlin dengan menggunakan Android Studio. Berikut adalah implementasi sistem Aplikasi Andalan Kita:

a. Aplikasi pengguna

1. Tampilan *Splash Screen*

Splash screen merupakan tampilan awal sebelum menu aplikasi terbuka. Tampilan splash screen dapat dilihat pada gambar berikut :



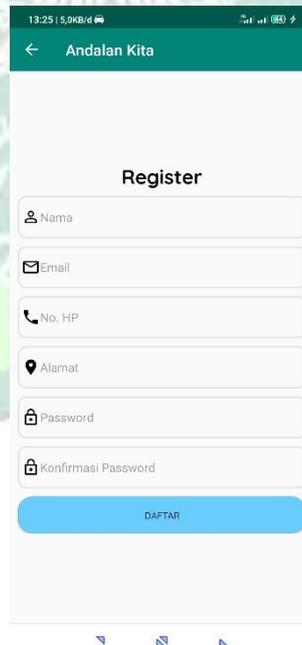
Gambar 4.31. Tampilan Splash Screen

2. Tampilan *Login/SignUp* User

Halaman *login/signup* merupakan halaman yang muncul pertama kali ketika aplikasi di jalankan. tampilan dari halaman login dapat dilihat pada gambar berikut :



Gambar 4.32. Tampilan Login



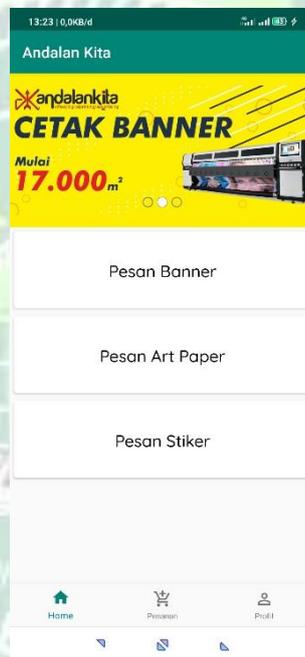
Gambar 4.33. Tampilan Signup

Penjelasan gambar :

Pertama user mengisi form login berupa email dan password akun yang telah terdaftar. Jika belum mempunyai akun, user memilih daftar disini pada halaman login dan mengisi form pendaftaran dan kembali ke halaman login untuk memasukan email dan password yang didaftarkan.

3. Tampilan Beranda

Pada tampilan beranda ini terdapat pilihan pemesanan produk. Tampilan menu ini dapat dilihat pada gambar berikut :



Gambar 4.34. Tampilan Beranda

4. Tampilan Pemesanan

Pada tampilan pemesanan user mengisi form pemesanan dan mengirim desain gambar. Tampilan pemesanan ini dapat di lihat pada gambar berikut :

13:24 | 0,1KB/d

← Pesan Banner

Nama Lengkap : Restu Agung Samekto

Nama Produk : Nama Produk

Ukuran : P XL

Jumlah Cetak :

Finishing : Lelebitan + Ring

Pilih Gambar

PESAN

Gambar 4.35. Tampilan Pemesanan

5. Tampilan Pesanan

Pada menu ini berisi daftar pesanan user yang sudah dikirim dan menampilkan informasi status pemesanan. Tampilan menu ini dapat dilihat pada gambar berikut :

13:24 | 0,5KB/d

← Andalan Kita

DAFTAR PESANAN BANNER

Restu Agung Samekto

Contoh

1Pcs

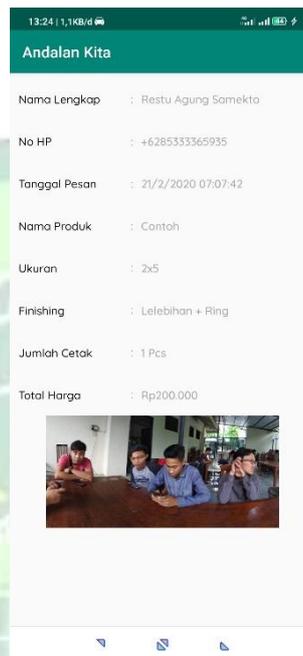
Rp200.000

Pesanan Sudah Jadi

Gambar 4.36. Tampilan Pesanan

6. Tampilan Detail Pesanan

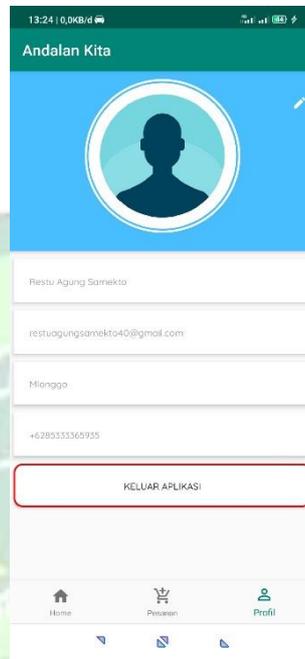
Pada tampilan ini merupakan tampilan penuh dari daftar pesanan user berupa deskripsi dan gambar yang telah dikirim. Tampilan pesanan ini dapat dilihat pada gambar berikut :



Gambar 4.37. Tampilan Detail Pesanan

7. Tampilan Profil

Pada menu profil menampilkan informasi dari akun user. Tampilan menu ini dapat dilihat pada gambar berikut:

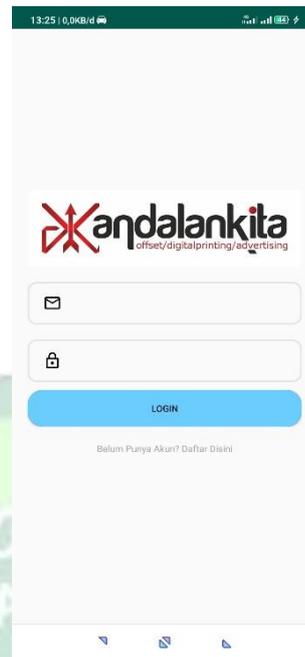


Gambar 4.38. Tampilan Menu Image User

b. Aplikasi Admin

1. Tampilan login admin

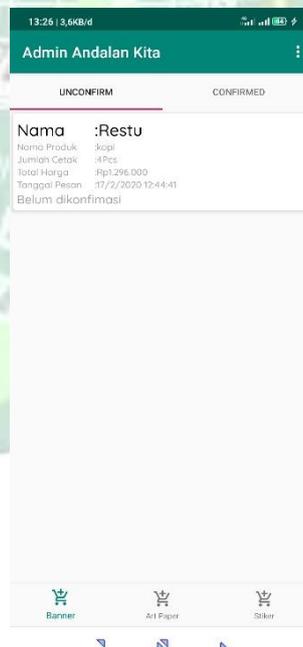
Pada halaman ini admin harus mengisi email dan password admin. Tampilan menu ini dapat dilihat pada gambar berikut :



Gambar 4.39. Tampilan Login Admin

2. Tampilan Melihat Pesanan

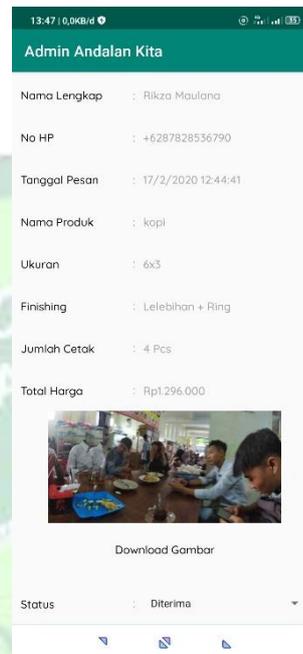
Pada menu ini admin memilih daftar pesanan dari user untuk dikonfirmasi. Tampilan menu chat dapat di lihat pada gambar berikut:



Gambar 4.40. Tampilan Melihat Pesanan

3. Tampilan Update Status Pesanan

Pada halaman ini admin melakukan konfirmasi informasi pemesanan dan mengunduh desain gambar dari user. Tampilan menu ini dapat dilihat pada gambar berikut :



Gambar 4.41. Tampilan Update Pesanan

4.2 Pengujian Metode

4.2.1 Black Box Testing

Pengujian metode pada aplikasi Andalan Kita menggunakan black box testing yang dilakukan pada seluruh tampilan aplikasi yang disajikan ke dalam table table pengujian sesuai fungsi fungsi dalam aplikasi.

1. Black Box Testing Pada Aplikasi

Berikut adalah table hasil pengujian *black box testing* untuk aplikasi Andalan Kita :

Tabel 4.3. Black Box Testing Pada Aplikasi

No.	Modul	Fungsi	Hasil yang diharapkan	Hasil pengujian
1.	Icon	Muncul pada device android	Muncul pada device android	Valid
2.	Splash screen	Tampilan ini menampilkan gambar logo aplikasi selama beberapa detik sebelum masuk ke halaman login	Dapat masuk ke halaman login	Valid
4.	Login	Menampilkan tombol login untuk masuk ke dalam menu utama	Dapat masuk ke menu utama untuk melihat daftar produk	Valid
5.	Daftar Produk	Merupakan bagian awal dari aplikasi setelah tampilan login	Menampilkan daftar pilihan pemesanan produk	Valid
6.	Halaman Pemesanan	Merupakan tampilan selanjutnya untuk daftar produk	Dapat melakukan pemesanan produk.	Valid
7.	Halaman Riwayat Pesanan	Merupakan bagian awal dari aplikasi setelah tampilan login	Menampilkan daftar riwayat pemesanan produk	Valid
8.	Halaman Pemesanan	Merupakan tampilan setelah daftar riwayat pemesanan	Dapat menampilkan informasi pemesanan	Valid
9.	Halaman Profil	Merupakan bagian awal dari aplikasi setelah tampilan login	Dapat menampilkan informasi data diri user	Valid
10.	Logout	Pada halaman ini menampilkan tombol keluar untuk keluar dari aplikasi	Dapat menampilkan halaman keluar dan akun user keluar dari aplikasi	Valid

4.3 Evaluasi Dan Hasil Validasi

4.3.1 Evaluasi System Aplikasi

Berdasarkan hasil pengujian fitur fitur halaman di aplikasi Andalan Kita dengan metode *black box testing* dari semua skenario

pengujian tiap tiap tabel hasil pengujian berjalan sesuai harapan. Jadi dapat disimpulkan dari pengujian system pada aplikasi Andalan Kita dengan *metode black box testing* adalah system berjalan sesuai dengan harapan dan valid.

4.3.2 Validasi Kelayakan Aplikasi

4.3.2.1 Validasi Ahli

1) Ahli Materi

Validasi yang dilakukan oleh ahli materi adalah mengumpulkan saran dan pendapat untuk melakukan revisi aplikasi Andalan Kita sebagai berikut :

Tabel 4.4. Hasil Penilaian Ahli Materi

No	Indikator	Jumlah Butir
1.	Apakah fungsi tombol <i>register</i> berfungsi dengan baik?	1
2.	Apakah tidak ada kendala saat mengisi <i>form register</i> ?	1
3.	Apakah fungsi tombol <i>login</i> berfungsi dengan baik?	1
4.	Apakah tidak ada kendala saat mengisi nama pemesan?	1
5.	Apakah hasil perhitungan ukuran produk sudah sesuai?	1
6.	Apakah fungsi tombol <i>upload</i> gambar berfungsi dengan baik?	1
7.	Apakah hasil perhitungan total biaya pemesanan produk sudah sesuai?	1
8.	Apakah fungsi status pemesanan produk sudah sesuai?	1
9.	Apakah menu edit profil berfungsi dengan baik?	1
10.	Apakah fungsi tombol <i>sigout</i> berfungsi dengan baik?	1
Jumlah		10
Jumlah Soal Pengujian		10
Jumlah Skor Ideal (N)		$3 \times 10 = 30$
Jumlah Nilai Pengujian (F)		30
Presentase Kelayakan (P)		$(30/30) \times 100 = 100\%$
Kriteria		Sangat Layak

2) Ahli Media

Validasi yang dilakukan oleh ahli media adalah mengumpulkan saran dan pendapat untuk melakukan revisi aplikasi Andalan Kita sebagai berikut :

Tabel 4.5. Hasil Penilaian Ahli Media

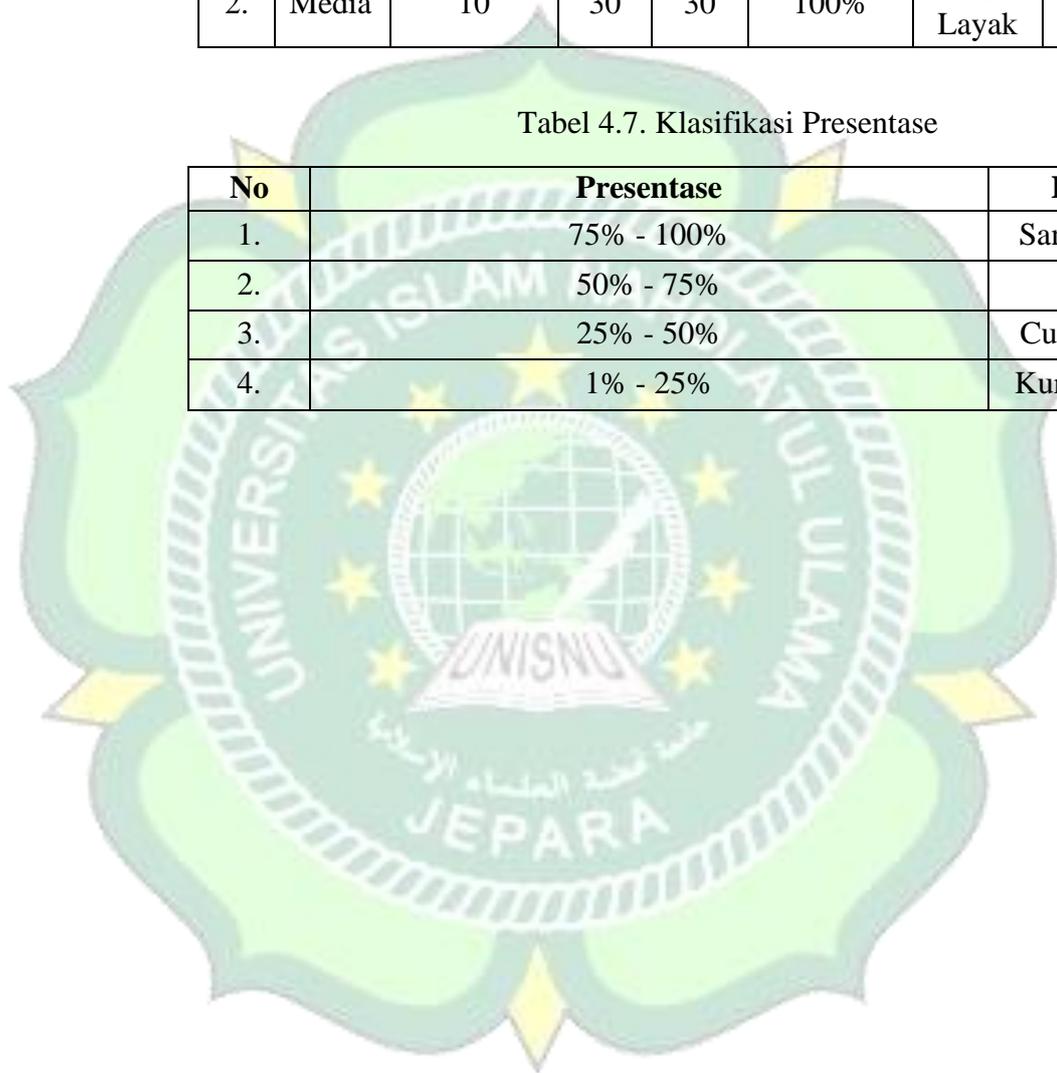
No	Indikator	Jumlah Butir
1.	Apakah perintah <i>sign up</i> atau mendaftar dalam aplikasi pemesanan produk berfungsi dengan baik?	1
2.	Apakah perintah <i>log in</i> atau masuk dalam aplikasi pemesanan produk berfungsi dengan baik?	1
3.	Apakah fitur mengisi nama pemesan dalam aplikasi pemesanan produk berfungsi dengan baik?	1
4.	Apakah fitur mengisi nama produk dalam aplikasi pemesanan produk berfungsi dengan baik?	1
5.	Apakah perintah <i>upload</i> atau mengunggah gambar dalam aplikasi pemesanan produk berfungsi dengan baik?	1
6.	Apakah fitur mengisi ukuran dalam aplikasi pemesanan produk berfungsi dengan baik?	1
7.	Apakah perintah pilihan <i>finishing</i> dalam aplikasi pemesanan produk berfungsi dengan baik?	1
8.	Apakah fitur mengisi jumlah produk dalam aplikasi pemesanan produk berfungsi dengan baik?	1
9.	Apakah fitur total biaya pemesanan dalam aplikasi pemesanan produk berfungsi dengan baik?	1
10.	Apakah perintah <i>log out</i> atau keluar dalam aplikasi pemesanan produk berfungsi dengan baik?	1
Jumlah soal pengujian		10
Jumlah skor ideal (n)		10x3 = 30
Jumlah nilai pengujian (f)		30
Presentase kelayakan (P)		$(30/30) \times 100 = 100\%$
Kriteria		Sangat Layak

Tabel 4.6. Validasi Ahli

No.	Ahli	Instrument	Skor Ideal (n)	Skor Total (f)	Presentase	Kriteria	Keterangan
1.	Materi	10	30	30	100%	Sangat Layak	-
2.	Media	10	30	30	100%	Sangat Layak	-

Tabel 4.7. Klasifikasi Presentase

No	Presentase	Kriteria
1.	75% - 100%	Sangat Layak
2.	50% - 75%	Layak
3.	25% - 50%	Cukup Layak
4.	1% - 25%	Kurang Layak



3) Hasil Angket Dari Responden

Penilaian aplikasi Andalan Kita juga dilakukan dengan menyebar angket yang berisi 8 butir pertanyaan kombinasi kepada 30 responden, berikut hasil rekapitulasi nilai sesuai butir pertanyaan :

Tabel 4.8. Skor Penilaian Oleh Responden Pelanggan

No	Indikator	Jumlah Butir
1.	Aplikasi pemesanan produk dapat meminimalisir antrian pemesanan produk di CV. Andalan kita.	1
2.	Aplikasi pemesanan produk menambah pengetahuan informasi pemesanan produk di CV. Andalan kita.	1
3.	Bahasa dan kalimat yang digunakan dalam aplikasi pemesanan produk di CV. Andalan kita mudah dipahami.	1
4.	Penggunaan jenis huruf, ukuran huruf, warna dan gambar dalam aplikasi pemesanan produk di CV. Andalan kita terlihat jelas.	1
5.	Aplikasi pemesanan produk di CV. Andalan kita ini mudah untuk dipelajari.	1
6.	Aplikasi pemesanan produk di CV. Andalan kita ini menarik.	1
7.	Aplikasi pemesanan ini mendorong rasa ingin tahu terhadap pemesanan produk di CV. Andalan kita.	1
8.	Aplikasi pemesanan produk di CV. Andalan kita ini dapat digunakan dimana saja dan kapan saja.	1
	Jumlah	1
	Jumlah soal pengujian	8
	Jumlah skor ideal (n)	$8 \times 5 \times 30 = 1200$
	Jumlah nilai pengujian (f)	1043
	Presentase kelayakan (P)	$(1043/1200) \times 100 = 86,92\%$
	Kriteria	Sangat layak

Tabel 4.9. Klasifikasi Presentase

No	Presentase	Kriteria
1.	75% - 100%	Sangat Layak
2.	50% - 75%	Layak
3.	25% - 50%	Cukup Layak
4.	1% - 25%	Kurang Layak

Tabel 4.10. Hasil Penilaian Angket Responden

No.	Jumlah Responden	Instrument	Skor Ideal (n)	Skor Total (f)	Presentase	Kriteria
1.	30	8	1200	1043	86,92%	Sangat Layak

