PENGGUNAAN *LIBRARY FAST ANDROID NETWORKING* UNTUK MEMBANGUN FITUR REGISTER PADA PROYEK LOTUS (STUDI KASUS: PT. AGUNG TRANS SOLUSINDO)

Andre Febrianto[1], Raras Franita[2]
Prodi Teknik Informatika, Jurusan Teknik Elektro Informatika dan Sistem Fisis
Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung – Lampung Selatan
e-mail: andre.febrianto@if.itera.ac.id[1]

ABSTRAKS
Lotus adalah sebuah aplikasi sosial media dan e-commerce berbasis android. Pada aplikasi Lotus terdapat modul-modul diantaranya adalah Lotus Connect. Lotus Connect merupakan aplikasi sosial media dimana pengguna dapat membagikan sebuah foto atau vidio, membagikan status berupa text, dan membagikan ulang sebuah postingan, serta berkomunikasi melalui pesan atau Direct Message.Pada pengembangan Lotus Connect penulis berkesempatan bergabung dalam team Android developer, yang membantu dalam pembuatan fitur register. Tools yang digunakan penulis antara lain Slack, Trello, Android Studio, PostMan, Gitlab. Penulis mempunyai role Sebagai frontend yang memanggil layanan dari back-end service dengan menggunakan library fast android networking serta dapat melakukan penyimpanan lokal dengan menggunakan Shared Preference.Pemilihan library sangat tergantung pada kebutuhan aplikasi. Pemilihan library fast android networking pada aplikasi lotus connect dikarenakan, featurs dan request type yang diberikan oleh library fast android networking sangat lengkap dan tepat untuk memanggil dari back-end service.

Kata Kunci: Networking, Preference,Lotus, Android Studio.

1. PENDAHULUAN

1.1 Latar Belakang

PT. Agung Trans Solusindo suatu perusahaan yang bergerak dalam bidang logistik yang beralamatkan di Jl. Trembesi Blok D Bandar Baru Komplek Kemayoran, Jakarta Utara. Perusahaan ini membuka cabang di Bandar Lampung, pada saat ini cabang yang berada di Bandar Lampung sedang mengerjakan proyek super app bernama LOTUS dengan fitur utama adalah layanan sosial media bernama *lotus connect,* layanan ini hampir sama dengan aplikasi sosial media pada umumnya, namun keunikan dari aplikasi lotus ini terdapat pada modul *lotus shop,* yaitu pengguna dapat bersosial media sambil berbelanja, aplikasi *lotus connect* dan *lotus shop* mempunyai sasaran pengguna masyarakat umum.

Lotus Connect merupakan aplikasi sosial media tidak berbayar yang dapat digunakan pada semua kalangan. Media sosial memberikan kesempatan bagi setiap orang untuk membuat pesan tanpa melihat latar belakang budaya, usia, status sosial, bahkan gender. Tidak asing lagi jika media sosial sangat diminati oleh kalangan remaja maupun dewasa, karena dengan kecanggihan atau power of aplication social media dapat mempercepat penyebaran informasi. Platform sosial media yang banyak digemari, yaitu instagram, twitter, facebook, dan tiktok. Semua aplikasi tersebut memiliki ciri khas tersendiri, semua orang dari berbagai penjuru dunia dapat mengakses aplikasi tersebut selama mempunyai koneksi internet.

Mudahnya penggunaan media sosial dalam bertukar informasi memberikan manfaat dari berbagai kepentingan, baik kepentingan sekelompok orang, instansi atau pribadi. Dari berbagai kepentingan tersebut, beberapa pihak menggunakan untuk menampilkan karyanya, kegiatan jual dan beli barang, atau hanya berbagi aktivitas kehidupan sehari-hari yang dapat dilakukan dengan mengunggah sebuah postingan. Hal inilah yang melatarbelakangi pembuatan aplikasi Lotus terkhusus pada layanan lotus connect.

1.2 Referensi

Lotus Connect merupakan sebuah aplikasi layanan sosial media tidak berbayar. Di dalam aplikasi ini pengguna dapat membagikan sebuah foto atau vidio, membagikan status berupa text, membagikan ulang sebuah postingan, serta berkomunikasi melalui pesan atau Direct Message, selama pengguna terhubung ke internet. Pengguna dapat mengakses aplikasi lotus connect dengan mendaftarkan akun e-mail atau nomor whatsapp. Setelah akun e-mail atau nomor whatsapp terverifikasi maka pengguna dapat memiliki akun pada aplikasi lotus yang akan digunakan untuk mengakses aplikasi tersebut.

FAN merupakan kepanjangan dari fast android networking adalah sebuah library yang dapat memudahkan kita untuk melakukan HttpConnectionRequest(codepolitan,2017). Dalam android ada berbagai macam opsi penyimpanan atau storage. Shared Preferences adalah salah satu penyimpanan yang digunakan dalam pengerjaan proyek Lotus(A Faturrahman, 2017).

Rudy Siahaan (2002:35). Email berbeda Android Studio adalah Integrated Development Enviroment (IDE) untuk sistem operasi Android, yang dibangun di atas perangkat lunak JetBrains

a. IntelliJ IDEA dan didesain khusus untuk pengembangan android dengan menggunakan bahasa pemrograman Kotlin(Developers, 2020).
b. Postman tools yang digunakan untuk melihat end-point atau API beserta response yang diberikan.
c. Figma sebagai alat untuk melihat design dari tim ui/ux designer
d. Git sebagai sistem pengontrol versi (Version Control System) pada proyek perangkat lunak[4]. Salah satunya adalah Gitlab adalah layanan yang menyediakan akses remote ke Git repositories(Yulianto, 2017) Git Bash adalah aplikasi untuk lingkungan Microsoft Windows yang menyediakan lapisan emulasi untuk pengalaman Git command line(Silvia, 2020).

Pada tools yang telah dijelaskan merupakan alat bantu yang digunakan penulis dalam pengembangan di sisi teknis pembuatan, dalam komunikasi antar tim terdapat tools yang digunakan penulis antara lain:

a. Trello sebagai manajemen task dan kalaborasi kerja tim, papan board yang terdapat backlog, active sprint, on going, dan done serta terdapat sprint baik yang aktif ataupun yang sudah lewat.
b. Slack sebagai alat komunikasi tim, tempat berdiskusi antar tim sedivisi dan juga tim seluruh tim developer.
c. Spreed Sheet sebagai catatan dari apa yang sudah dilakukan (done) dan yang akan dilakukan (to-do). Setiap hari pada jam 11 terdapat daily stand up yaitu memberikan progress atau task yang telah dikerjakan.

2. METODE PENELITIAN

2.1 Analisis Kebutuhan Sistem

Fast Android Networking merupakan library terpopuler yang digunakan pada semua jenis jaringan yang memanggil layanan dari *back-end service* di aplikasi Android. FAN dibuat diatas lapisan jaringan *OkHttp*, *library* FAN menangani setiap jenis permintaan dan menunggu tanggapan atau *response* dari server[7]. Adapun beberapa fungsi dari FAN antara lain sebagai berikut:

- Mengunduh semua jenis file
- Mengunggah semua jenis file
- Membatalkan permintaan dukungan
- Mengatur prioritas untuk permintaan apapun

Ada banyak *library* yang tersedia diantaranya *Retrofit, Volley and Ok Http*. Pada umumnya *developer* membutuhkan *library* untuk pertukaran data, sebagai contoh unduh-unggah gambar, file media seperti audio, vidio dan jenis file lainnya. Pengembang memilih *library fast android networking* dikarenakam *library* ini memenuhi semua kebutuhan pada proyek lotus. Beberapa perbedaan fitur yang disediakan oleh *Library Fast Android Networking.*

Library FAN mengirim data berupa “GET” , “POST” atau upload gambar ke internet serta dapat mengambil data dari internet. Method GET menampilkan data dan menambahkan URI yang nantinya ditampung pada action(A Mukhtar, 2020). Method POST mengirimkan data langsung kepada action tanpa melalui URI sehingga bersifat uncacheable (tidak dapat disimpan pada cache) (A Mukhtar, 2020). FAN lebih unggul daripada library volley dalam hal mengirim gambar ke internet atau web, dibuktikan pada tabel 1.

Tabel 1. Fitur yang didukung beberapa *library*[9]

Featurs	*Http Client*	*Volley*	*Retrofit*	*Fast Android Networking*
POST	Yes	Yes	Yes	Yes
Multipart	Yes	Yes	Yes	Yes
Different Request Type	No	Yes	Yes	Yes
JSON Request	No	Yes	Yes	Yes
Priorities	No	Yes	No	Yes
Multiple Request	No	Yes	No	Yes
Caching	No	Yes	No	No
Image Loading	No	Yes	No	Yes
Retry Mechanism	No	Yes	Manual	No
Manual Request Concellation	No	Yes	Yes	Yes

Pada tabel 1 terdapat library Http-Client, Volley, Retrofit dan juga Fast Android Networking mengenai tabel diatas menjelaskan fitur-fitur yang didukung dari sebuah library. Pada library Volley mendukung semua fitur sedangkan pada library Fast Android Networking fitur caching dan retry mechanism tidak didukung oleh FAN. Penulis lebih memilih menggunakan FAN dikarenaka type request yang diberikan oleh FAN lebih lengkap dibandingkan volley yang dibuktikan pada tabel 2.

Tabel 2. *Type Request form library*[9]

Volley	*Retrofit*	*Fast Android Networking*
JSON Object Request	*String*	*Object*
JSON Array Request	*Object*	*Object*
String Request	*Collection*	*Collection*
Image Request	*Boolean*	*Boolean*
-	*Integer*	*Integer*
-	*Date*	*Date*
-	-	*Image*

P-ISSN:0216-9436
E-ISSN:2622-6782

-	-	*File*

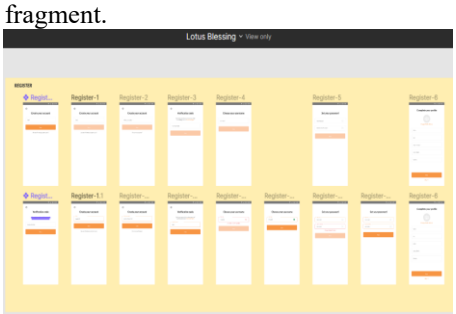
Pada tabel 2 *library* FAN memiliki *type request image* dan *file* yang tidak dimiliki *volley* dan *retrofit*, sehingga pada pengembangan aplikasi lotus terdapat unduh-unggah sebuah file seperti *image, audio,* dan vidio. *Type request* yang disediakan dan didukung oleh FAN memenuhi kebutuhan aplikasi lotus.

Di android ada berbagai macam opsi penyimpanan atau *storage. Shared Preferences* merupakan salah satu penyimpanan. Karakter penyimpanannya adalah *key-value storage,* sehingga hanya bisa menyimpan data bertipe primitif (float, int, longm string, boolean) (A Faturrahman, 2017). Penyimpanan ini tergolong minimum dan sangat cocok untuk penyimpanan data yang sedikit, oleh karena itu pembuatan *register* dari aplikasi Lotus mengunakan *Shared Preference.*

2.2 Proses Pelaksanaan

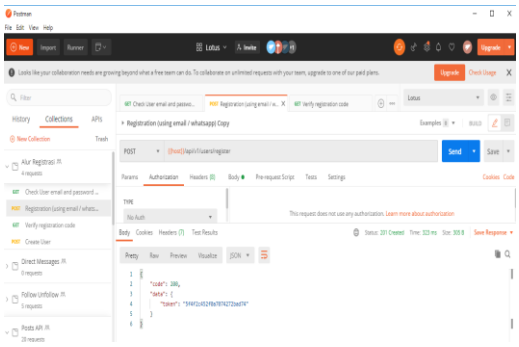
Untuk membangun fitur register dimulai dari tahap perencanaan dimana bekerja sama dengan tim UI/UX designer untuk membuat interface dari design yang telah dibuat, lalu berdiskusi dengan tim back-end untuk mengetahui response yang diberikan dari service yang telah dibuat. Setelah design dari tim UI/UX designer selesai dan service dari tim back-end sudah selesai maka dimulai tahap implementasi.

1. Tahap awal yaitu dengan membuat interface dari tim UI/UX designer, setelah selesai maka mengimpelemtasikan Android Jetpack dan Android Architecture Components (Arch Component), salah satunya ialah komunikasi antar fragment.



Gambar 1. Tampilan design fitur *register* dari tim UI/UX Designer

2. Dilanjutkan dengan memanggil service dari back-end, yaitu merupakan otak dari jalannya sebuah program dengan menggunakan library fast android networking.



Gambar 2. Tampilan *service* dari tim *Back-end*

3. Setelah itu data user yang telah diinputkan pada saat register disimpan dalam penyimpanan lokal dengan menggunakan Shared Preference, dan disimpan dalam storage dalam class sharedpref manager pada gambar 3.

```
val token: Token
    get() {
        val sharedPreferences : SharedPreferences =
            mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE)
        return Token(
            sharedPreferences.getString("token", null)
        )
    }
val user: User
    get() {
        val sharedPreferences : SharedPreferences =
            mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE)
        return User(
            sharedPreferences.getInt("__v", 0),
            sharedPreferences.getString("_id", "wkwk"),
            sharedPreferences.getString("avatar", null),
            sharedPreferences.getString("bio", null),
            sharedPreferences.getBoolean("createdAt", false),
            sharedPreferences.getBoolean("deleted", false),
            sharedPreferences.getString("email", null),
            sharedPreferences.getBoolean("emailVerified", false),
            sharedPreferences.getString("name", null),
            sharedPreferences.getString("password", null),
            sharedPreferences.getString("phone", null),
            sharedPreferences.getInt("postsCount", 0),
            sharedPreferences.getString("updatedAt", null),
            sharedPreferences.getString("username", null)
        )
    }
```

Gambar 3. Penyimpanan di lokal

```
val user : User = SharedPrefManager.getInstance(requireContext()).user
val updateUser = User( __v: 0, _id: null, avatar: null, bio: null, createdAt: false, deleted: false, user.email,
    emailVerified: false, name: null, pw2.text.toString(), user.phone, postsCount: 0, updatedAt: null, user.username)
SharedPrefManager.getInstance(requireContext()).saveUser(updateUser)
```

Gambar 4. Cara penyimpanan data di *local.*

4. Penyimpanan state, ketika aplikasi mengalami for close maka masih tersimpan state terakhir, sehingga pengguna dapat melanjutkan register tidak perlu mengulang registrasi dari awal.

```
//for save state
fun saveState(state: String) {
    val sharedPreferences : SharedPreferences = mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE)
    val editor : SharedPreferences.Editor = sharedPreferences.edit()
    editor.putString("state", state)
    editor.apply()
}

// for get state
val state: String
    get() {
        val sharedPreferences : SharedPreferences =
            mCtx.getSharedPreferences(SHARED_PREF_NAME, Context.MODE_PRIVATE)
        return sharedPreferences.getString("state", null).toString()
    }
```

Gambar 5. Penyimpanan state di lokal

2.3 Tahapan Pekerjaan

Tahapan atau alur secara garis besar dari fitur register dengan mendaftarkan akun e-mail atau nomor whatsapp.

P-ISSN:0216-9436
E-ISSN:2622-6782

Gambar 6. Tahapan *Register* Pada Aplikasi Lotus

Pada gambar 6 merupakan alur atau jalannya proses *register* yang bertujuan untuk mendapatkan akun pada aplikasi lotus. Berikut tahapan secara garis besar di fitur *register* :

1. Pertama *user* bisa memilih untuk mendaftarkan akun *e-mail* atau nomor *whatsapp*. Jika akun *e-mail* atau no *whatsapp* terverifikasi maka masuk pada tampilan *Verification Code*.
2. Lalu sistem akan mengirimkan sebuah *code* verifikasi pada akun *e-mail* atau nomor *whatsapp* yang telah didaftarkan.
3. Jika sudah menerima *code* maka pengguna akan menginputkan *code* tersebut untuk melanjutkan tahap registrasi. Jika pengguna tidak menerima code verifikasi, terdapat fitur untuk *resend code*.
4. Ketika *code* sudah terverifikasi, lanjut ke tampilan *username* dimana pengguna menginputkan *username*.
5. Jika *username* telah terverifikasi maka akan lanjut pada tampilan *password* dimana pengguna menginputkan *password*.
6. Tahap terakhir ialah pengisian data profil dimana pengguna dapat menambahkan informasi tambahan tentang data diri dan pengguna juga dapat melewatinya.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dari pembuatan fitur register pada proyek lotus adalah

1. Pada gambar 7 merupakan tampilan awal aplikasi lotus, jika pengguna belum memiliki akun maka dapat membuat akun pada fitur *register* di paling bawah aplikasi.

Gambar 7. Halaman atau tampilan awal aplikasi Lotus

2. Setelah pengguna masuk ke fitur *register* maka muncul membuat akun dengan menggunakan *e-mail*, jika ingin menggunakan *whatsapp* maka mengklik ”*use your whatsapp number”,* maka pengguna dapat mendaftar menggunakan nomor *whatsapp.*

Gambar 8. Register with E-mail

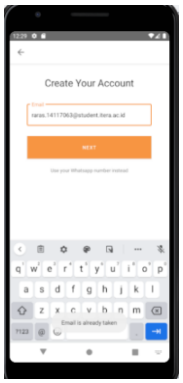
Gambar 9. *Register with whatsapp number*

P-ISSN:0216-9436
E-ISSN:2622-6782

3. Pengguna menginputkan akun *e-mail* lalu sistem akan mengecek apakah akun email yang diinputkan respon status *available* atau *no available* dengan membuat fungsi pada gambar 10 yang menggunakan *library fast android networking*.

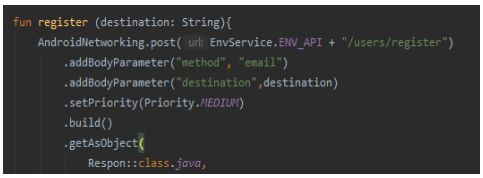
Gambar 10. Fungsi *Check Email*

Pada gambar 11 merupakan fungsi cek *e-mail*, menggunakan *AndroidNetworking.**get*** *(end-point dari backend service*). Sistem dari *backend service* akan mengecek apakah status dari *e-mail* yang di inputkan oleh *user* sudah terdaftar atau belum. Jika *e-mail* sudah terdaftar maka, akan memberikan pemberitahuan ke pengguna bahwa ”*email is already taken”*.



Gambar 11. Akun email sudah digunakan

4. Ketika *E-mail* status *available* maka akan memanggil fungsi *register*, pada gambar 12 merupakan fungsi *register* untuk mengirimkan code verifikasi dengan menggunakan *Android-Networking**post***(*end-point* dari *backend service*). Sistem dari *back-end* akan mengirimkan *code* verifikasi dengan methode *e-mail* atau nomor *whatsapp* dan *destination* merupakan yang di inputkan oleh pengguna.



Gambar 12. Fungsi *register* mengirimkan code verifikasi

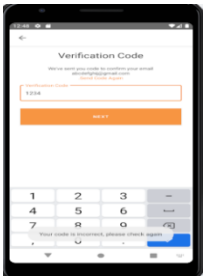
Kemudian sistem akan mengirimkan kode verifikasi ke akun *e-mail* atau nomor *whatsapp* seperti pada gambar 13.

Gambar 13. *Code* verifikasi

5. Setelah mendapat kode verifikasi, pengguna akan menginputkan *code*. Sistem akan mengecek apakah *code* yang diinputkan *user* benar atau salah dengan menggunakan fungsi *check code*.

Gambar 14. Fungsi check code

Pada Gambar 14 fungsi memanggil dari *back-end service* yang akan mengecek *code* dari inputan *user* dan juga token yang didapat saat mendaftarkan akun *e-mail* atau nomor *whatsapp*. Jika inputan atau token yang diberikan ke *server* tidak sesuai, maka pengguna akan mendapat pesan *code incorrect*.

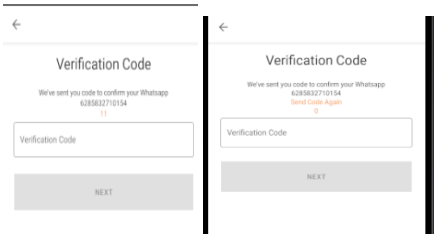


Gambar 15. *Code* verifikasi yang diinputkan tidak sesuai

6. Pada gambar 15, jika *code* verifikasi yang diinputkan salah maka pengguna dapat memilih

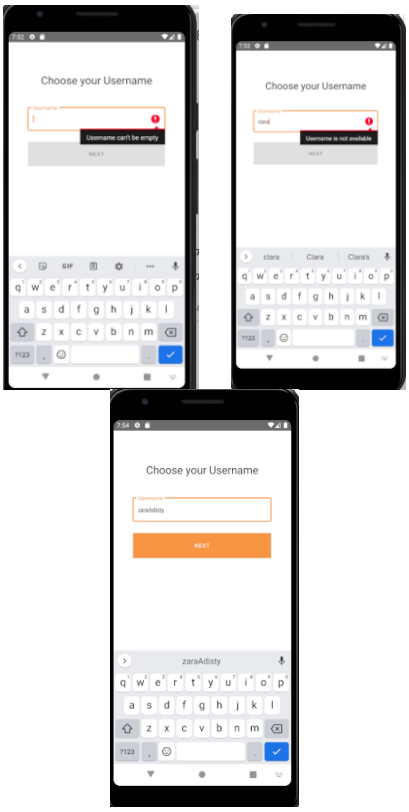
P-ISSN:0216-9436
E-ISSN:2622-6782

fitur *resend code*, selain itu *resend code* digunakan saat pengguna tidak mendapatkan *code* verifikasi.



Gambar 16 *Resend code*

7. Ketika sudah mendapat respon 200 yang berarti *code* dari inputan user telah sesuai dan juga token *match* maka tahap selanjutnya adalah *validate username*. Adapun ketentuan pengisian *username*, yaitu username tidak boleh kosong ,tidak boleh memakai angka, simbol dan minimal 6 *character* jika tidak sistem akan memberikan *set error* seperti gambar 17 jika pengguna menginputkan yang tidak sesuai.



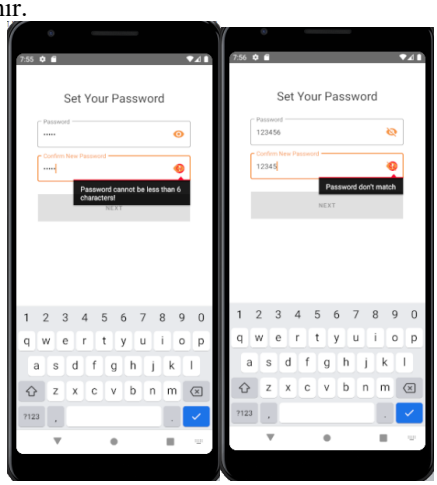
Gambar 17. *Validate Username*

Jika *user* sudah benar maka akan *button next* akan aktif dan bisa melanjutkan ketahap selanjutnya namun terdapat pengecekan *username*, yaitu *username* sudah ada di DB atau tidak, dengan menggunakan fungsi cek *username* pada gambar 18 jika *code* respon 200 yang berarti *username* belum ada di *database*. Hal ini bertujuan agar tidak terjadi duplikasi *username*.

```
fun checkUsername(username: String) {
    AndroidNetworking.get(EnvService.ENV_API + "/users/check?username=$username")
        .setPriority(Priority.MEDIUM)
        .build()
        .getAsObject(
            Respon::class.java,
            object : ParsedRequestListener<Respon> {
                override fun onResponse(respon: Respon) {
                    val gson = Gson()
                    if (respon.code.toString() == "200") {
                        val dataJson :String! = gson.toJson(respon.data)
                        val data :Data! = gson.fromJson(dataJson, Data::class.java)
                        Log.d( tag: "Cek Username",  msg: "Success")
                        Log.d( tag: "Status", data.status.toString())
                        if (data.status.toString() == "AVAILABLE") {
                            val dataJson :String! = gson.toJson(respon.data)
```

Gambar 18. Fungsi cek *username*

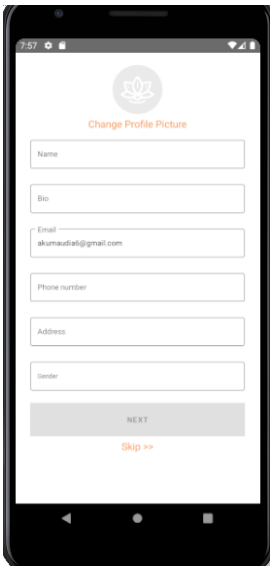
8. Setelah berhasil pengguna menginputkan *username* maka tahap selanjutnya adalah menginputkan *password*. Validate *password* minimal 6 karakter dan password harus match. Jika inputan *user* sudah benar maka tombol *button* dapat di klik.dan melanjutkan tahap akhir.



Gambar 19. *Password*

9. Setelah berhasil menginputkan *password* maka masuk pada tahap akhir yaitu menginputkan data profil. Pengguna dapat memilih *skip* atau memilih melengkapi data profil .

P-ISSN:0216-9436
E-ISSN:2622-6782



Gambar 20. Data *Profile*

Pada gambar 20 merupakan fungsi *create user* setelah data-data yang telah diinputkan oleh user dan disimpan di lokal maka data tersebut akan digunkan untuk membuat akun dengan menggunakan fungsi dengan *library fast android networking* ***upload*** seperti pada gambar 21.

```
private fun createUser(v: View, name: String) {
    val emailInProfile: EditText? = v.findViewById(R.id.emailProfile)
    val regisEmail :String?  = SharedPrefManager.getInstance(requireContext()).user.email
    val regisPhone :String?  = SharedPrefManager.getInstance(requireContext()).user.phone
    val user :User  = SharedPrefManager.getInstance(requireContext()).user

    val createUser : ANRequest.MultiPartBuilder<(raw) ANRequest.MultiPartBuilder<*>!>!
            = AndroidNetworking.upload( url: EnvService.ENV_API + "/users/create")
    val avatar :File?  = if (profilePicture != null) {
        File(profilePicture!!.path)
    } else {
        null
    }

    if (name == "") {
        createUser
            .addMultipartParameter("name", etNamePP.text.toString())
    } else {
        createUser
            .addMultipartParameter("name", name)
    }
```

Gambar 21. Mendaftarkan akun

```
//check register with email or phone
if (regisEmail != "") {
    createUser
        .addMultipartParameter("email", user.email)
        .addMultipartParameter("phone", etPhoneNumber.text.toString())
} else if (regisPhone != "" && emailInProfile?.text.toString() != "") {
    createUser
        .addMultipartParameter("phone", user.phone)
        .addMultipartParameter("email", emailInProfile?.text.toString())
    Log.d( tag: "emaildummy2", emailInProfile?.text.toString())
} else {
    createUser
        .addMultipartParameter("phone", user.phone)
        .addMultipartParameter("email", "asd@gmail.com")
    Log.d( tag: "emaildummy3", emailInProfile?.text.toString())
}

createUser
    .addMultipartFile( key: "profilePicture", avatar)
    .addMultipartParameter("wa", user.phone)
    .addMultipartParameter("username", user.username)
    .addMultipartParameter("password", user.password)
    .addMultipartParameter("bio", etBio.text.toString())
    .addMultipartParameter("gender", etGender.text.toString())
    .addMultipartParameter("address", etAddress.text.toString())
    .setTag(context)
    .setPriority(Priority.HIGH)
    .build()
```

Gambar 22. Pengecekan

4. KESIMPULAN

Kesimpulan yang dapat diperoleh dari laporan penelitian ini adalah:

1. Pembuatan fitur *register* pada proyek Lotus dilakukan secara bertahap mulai dari pengenalan proyek, pembuatan tampilan, pembuatan fungsi, hingga *testing.*
2. Fitur *register* dibuat dengan menggunakan *library fast android networking* untuk memanggil *service* dari *back-end*.

PUSTAKA

CodePolitan, “Library yang Wajib Kamu Coba Untuk Membuat Aplikasi Android”, www.codepolitan.com,2017. https://www.codepolitan.com/library-yang-wajib-kamu-coba-untuk-membuat-aplikasi-android-59b254b6d153c (accessed Aug. 27, 2020).

A. Fatturahman, “Belajar Membuat Aplikasi dengan Shared Preferences AndroidStudio”, www.okedroid.com,2017. https://www.okedroid.com/2017/08/belajar-membuat-aplikasi-dengan-sharedpreferences-android-studio.html (accessed Aug. 27, 2020).

developers, “Mengenal Android Studio,”www.developer.android.comhttps://developer.android.com/studio/intro?hl=id (accessed Aug. 28, 2020).

idcloudhost, “Pengertian dan Manfaat GIT bagi Developer,” https://idcloudhost.com, 2016. https://idcloudhost.com/pengertian-dan-manfaat-git-bagi-developer/ (accessed Aug. 30, 2020).

R. Yulianto, “Gitlab, Layanan Penyimpan Git gratis dan Open Source”, www.codepolitan.com, 2016.https://www.codepolitan.com/gitlab-layanan-penyimpan-git-gratis-dan-open-source (accessed Aug. 31, 2020).

Silvia, “Apa Itu Git Bash,” www.jetorbit.com, 2020. https://www.jetorbit.com/blog/apa-itu-git-bash/ (accessed Aug. 31, 2020).

A. Mukhtar, “Fast Android Networking merupakan library terpopuler”, https://medium.com, 2018.https://medium.com/@azammukhtar3/crud-mysql-android-dengan-fast-android-networking-fan-create-part-1-5ed8e2f893de (accessed Sep. 01, 2020).

A. Amelia, “Pengertian dari Metode Get dan Post,” https://ayunovitaa.blogspot.com, 2012.https://ayunovitaa.blogspot.com/2012/11/pengertian-dari-metode-get-dan-post.html (accessed Sep. 01, 2020).

J. Bhade, “Evaluation of Android Networking Libraries,” Int. J. Sci. Res. Eng. Trends, vol. 5, no. 4, pp. 1374–1377, 2019.