SISTEM PENGUNCI APLIKASI MENGGUNAKAN *TIMER* BERBASIS ANDROID

Wahyu Purnama Sari[1], Yiyi Supendi[2], Deden Wahyudi Julianto[3]
Program Studi Teknik Informatika[1,2,3]
Universitas Langlangbuana[1,2,3]
wahyu.alyapurnama@gmail.com[1], yiyi.supendi@gmail.com[2], dedenwahyudi101@gmail.com[3]

Abstrak

Perkembangan jaman dewasa ini telah menempatkan teknologi informasi sebagai layanan yang tidak terpisahkan dari kehidupan sehari-hari manusia. Perangkat komunikasi *smartphone* hampir dimiliki oleh seluruh penduduk dunia. Dunia dalam genggaman, itulah realita hidup manusia saat ini. Segala kegiatan dalam berbagai sektor mulai dialihkan dengan pemanfaatan aplikasi untuk mempermudah kerja manusia. Aplikasi-aplikasi berkembang setiap harinya, mulai aplikasi hiburan, pendidikan, perbankan, niaga/*e-commerce*, media sosial dan yang lainnya. Semua aplikasi ini dapat diunduh melalui layanan playstore pada platform Android. Pada penggunaan *smartphone*, telah banyak dikembangkan aplikasi pengunci untuk mengamankan data dari akses pihak-pihak yang tidak bertanggungjawab. Aplikasi pengunci ini biasanya digunakan pada saat seseorang memulai untuk membuka *smartphone* yang dimiliki. Sehingga ketika seseorang telah berhasil masuk ke *system*, maka akan mudah untuk membuka aplikasi-aplikasi lainnya dalam *smartphone* tersebut. Sistem pengunci aplikasi dengan menggunakan *timer* akan membantu seseorang untuk melindungi data-data dalam aplikasi yang terinstal dalam platform android. Aplikasi tertentu dapat dikunci menggunakan durasi waktu yang telah ditentukan. Selama durasi penguncian aplikasi berlangsung, maka *interface* aplikasi tersebut akan disembunyikan, sehingga tidak dapat terlihat oleh user yang menggunakannya. Aspek yang dilindungi mencakup *Availability*, *Integrity* dan *Confidentiality*.
Kata kunci: *Smartphone*, Android, Sistem Pengunci Aplikasi, *Timer*

Abstract

The development of today's era has placed information technology as an inseparable service from human daily life. Smartphone communication devices are almost owned by the entire world population. The world is in hand, that is the reality of human life today. All activities in various sectors began to be shifted to the use of applications to facilitate human work. Applications *are develo*p*ing every day, ranging from entertainment applications, education, banking, commerce/e-commerce, social media and others. All of these applications can be downloaded through the playstore service on the Android platform. In the use of smartphones, many locking applications have been developed to secure data from access by irresponsible parties. This lock application is usually used when someone starts to open their smartphone. So that when someone has successfully entered the system, it will be easy to open other applications on the smartphone. The application lock system using a timer will help someone to protect the data in applications installed on the android platform. Certain apps can be locked using a predefined time duration. For the duration of the application lock, the application interface will be hidden, so that it cannot be seen by the user who uses it. Aspects that are protected include Availability, Integrity and Confidentiality.*
Keywords: Smartphone, Android, App Lock System, Timer

I. PENDAHULUAN

Penggunaan *smartphone* oleh sebagian besar masyarakat masih belum memperhatikan aspek-aspek keamanan. Keamanan data-data yang tersimpan dalam *smartphone* harus dilindungi dari akses pihak-pihak yang tidak bertanggungjawab. Kelalaian dalam mengamankan data-data privasi dapat berkakibat fatal terhadap hilangnya data-data penting, modifikasi dan pemalsuan. Sesekali terkadang *smartphone* tergeletak di sembarang tempat di rumah atau di kendaraan pribadi. Tentunya ini akan menjadi sasaran anak-anak untuk menggunakannya tanpa pengawasan. Atau mungkin bisa saja *smartphone* orangtua yang digunakan anak-anak untuk belajar dalam jaringan (Daring). Pengamanan aplikasi berupa *Personal Identification Number* (PIN), Pola *Password*, kata sandi dan lain-lain merupakan langkah awal untuk melindungi *system* dari akses pihak lain. Ketika seseorang berhasil mengetahui PIN/Pola/Sandi *Smartphone*, maka akan mudah untuk membuka semua aplikasi yang terinstal dalam *smartphone* tersebut, karena semua aplikasi terpampang dalam platfoem yang digunakan. Sistem pengunci aplikasi dikembangkan dengan tujuan untuk menyembunyikan aplikasi-aplikasi tertentu dalam *smartphone* dari pandangan user. Dengan durasi yang telah diatur, dan aplikasi yang dipilih, maka aplikasi ini akan dikunci oleh *system*. Selama aplikasi disembunyikan, tampilan/*interface* akan dialihkan ke notifikasi berupa gambar tanpa menggangu layanan aplikasi lainnya. Hal ini tentunya akan membantu bagi mereka yang masih harus berbagi penggunaan *Smartphone* secara bergantian dengan anak-anak. Aplikasi tertentu dapat disembunyikan selama *smartphone* tersebut digunakan oleh pihak lain (anak-anak).

II. TINJAUAN PUSTAKA

1. *Smartphone*

Smartphone adalah telepon genggam yang memiliki fitur kegunaan yang berfungsi yang menyerupai komputer. Sampai saat ini belum ada standar pabrik yang menentukan arti dari *smartphone*. Bagi beberapa orang *smartphone* merupakan telepon genggam yang bekerja menggunakan semua perangkat lunak sistem operasi yang memiliki hubungan standar dan mendasar bagi pengembang aplikasi[1].

Sistem Android adalah sebuah *system* operasi untuk perangkat bergerak berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka untuk agar para pengembang untuk menciptakan aplikasi mereka sendiri. Pada awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru dalam pengembangan piranti lunak untuk ponsel atau *smartphone*[2].

Android mempunyai berbagai keunggulan sebagai software yang mengunakan basis kode komputer yang bisa dibagikan secara terbuka (*open source*) sehingga pengguna dapat dengan mudahnya membuat aplikasi baru di dalamnya. Android mempunyai aplikasi *native* Google yang terintegrasi seperti *pushmail Gmail, Google Maps, dan Google Calendar*[3].

Android Studio merupakan lingkungan terpadu atau *Integrated Development Environment* (IDE) yang resmi untuk pengembangan aplikasi *Android*, yang didasarkan kepada IntelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang handal, *Android Studio* juga menawarkan beberapa fitur-fitur yang dapat meningkatkan produktivitas Anda saat membuat suatu aplikasi Android, keuntungan bisa dilihat seperti dibawah ini: [4]

a. Emulator yang cepat dan memiliki berbagai fitur
b. Lingkungan terpadu tempat untuk bisa mengembangkan aplikasi untuk semua perangkat Android
c. Terapkan Perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi atau bisa dikenal *auto save*
d. Template kode dan integrasi GitHub untuk membantu Anda membuat fitur-fitur yang mempunyai banyak fungsi untuk aplikasi umum dan mengimpor beberapa kode untuk sampel
e. *Framework* dan alat pengujian yang lengkap
f. Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
g. Dukungan dari C++ dan NDK
h. Dukungan bawaan untuk produk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan *App Engine*
i. Sistem build berbasis *Gradle* yang fleksibel Android Studio merupakan *Integrated Development Environment* (IDE) yang jika diartikan menjadi Lingkungan Pengembangan Terpadu *official* yang dikeluarkan oleh Google untuk Android. IDE yang pertama kali rilis tahun 2014 ini mempunyai banyak sekali fungsi yang perlukan oleh para developer ketika pengembangan aplikasi Android seperti melakukan *build, test, debug*, dan lain sebagainya.

2. *Rapid Application Development (RAD)*

Terdapat tiga fase dalam *Rapid Aplication Development* (RAD) yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. Adapun ketiga fase tersebut adalah *requirements planning* (perencanaan syarat-syarat), RAD *design workshop* (*workshop* desain RAD), dan *implementation* (implementasi). Gambar 1 merupakan siklus model RAD[5].

Gambar 1. Siklus model RAD [5]

Adapun tahapan model RAD dapat dijelaskan sebagai berikut:

a. *Requirements Planning* (Perencanaan Syarat-Syarat) Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasikan tujuan-tujuan aplikasi atau sistem serta untuk megidentifikasikan syarat-syarat informasi yang ditimbulkan dari tujuantujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.
b. *Workshop* Desain RAD Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna.

c. *Implementation* (Implementasi) Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan *merancang* aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan *system*-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi.

3. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah bahasa yang digunakan untuk menentukan, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan pada proses pembuatan perangkat lunak, biasanya *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan *system* non-perangkat lunak lainnya. Selain itu UML merupakan Bahasa pemodelan yang menggunakan konsep orientasi *object*.

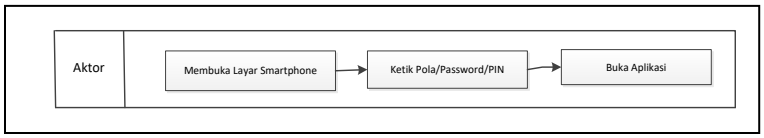
UML merupakan bahasa visual saat pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung". Beberapa permodelan yang termasuk kepada pemodelan UML seperti *use case diagram, class diagram, activity diagram*, dan *sequence diagram*.

III. ANALISIS DAN PERANCANGAN

Analisis merupakan aktivitas yang dilakukan untuk mendefinisikan proses yang terjadi pada sistem saat ini untuk kemudian diusulkan sistem yang akan di bangun. Proses bisnis digunakan untuk mempermudah pendeskripsian alur sistem yang terjadi. Analisis proses bisnis ini bertujuan untuk menguraikan proses secara sistematis *system* pengunci yanga akan dikembangkan

1. Analisis Sistem Saat Ini.

Analisis ini menggambarkan tentang prosedur mengunci aplikasi. Gambaran proses bisnis saat ini dapat dilihat pada gambar 2.

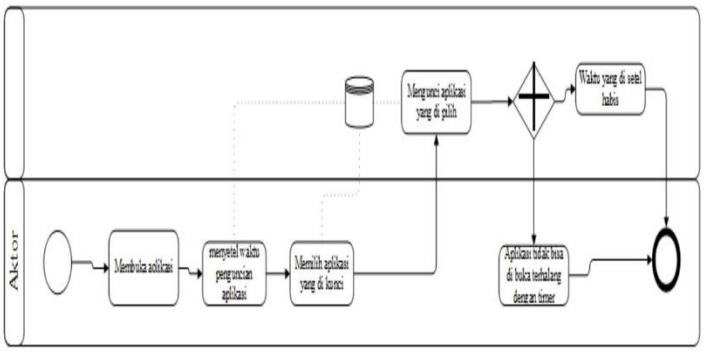


Gambar 2. Proses Bisnis Saat Ini

Proses bisnis yang terjadi saat ini hanya merupakan prosedur penguncian aplikasi menggunakan PIN/Pola/*Password* atau kata sandi pada saat akan membuka *smartphone*

2. Analisis Sistem Yang Diusulkan

Analisis sistem yang diusulkan menggambarkan tentang prosedur sistem pengunci aplikasi yang diusulkan. Adapun gambaran proses bisnis dapat dilihat pada gambar 3

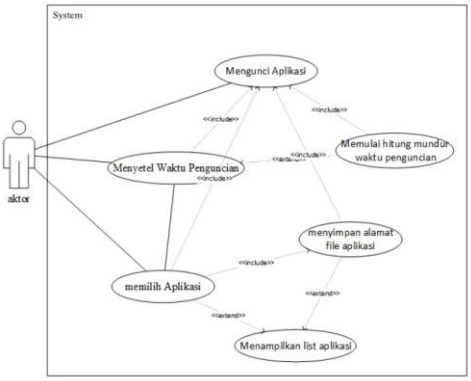


Gambar 3. Proses Bisnis yang Diusulkan

Pada gambar 3 dijelaskan ketika user membuka aplikasi, user akan diarahkan untuk mengatur waktu punguncian aplikasi kemudian memilihnya dan mulai melakukan penguncian. Selama rentang waktu penguncian, *interface icon* aplikasi yang dipilih akan disembunyikan sampai *timer* yang telah di atur selesai.

3. *Use Case Diagram*

Use Case digunakan untuk mengetahui prosedur *system* penguncian aplikasi.



Gambar 4. *Use Case Diagram*

Adapun untuk scenario *Use Case Diagram* dijelaskan pada tabel 1 samapi dengan tabel 3 berikut: Skenario *Use Case Diagram* Aplikasi dijelaskan pada tabel 1.

TABEL I
SKENARIO USE CASE MENYETEL WAKTU PENGUNCIAN

Use case Login	
Aktor	Pengguna aplikasi
Tujuan	Menyetel waktu penguncian
Kondisi Awal	User membuka aplikasi
Skenario	
Aksi Aktor	**Reaksi Sistem**
1. Aktor membuka aplikasi applock	
	2. Memeriksa waktu terakhir penguncian
	3. Menampilkan halaman *timer*.
4. Memasukan lama waktu penguncian aplikasi	
	5. Sistem menyimpan lama penguncian aplikasi dengan satuan menit
6. Aktor mejalankan waktu yang telah di tentukan dengan menekan tombol *start*	
	7. Sistem menjalankan hitung mundur
	8. Sistem membuka halamanpengingat agar aplikasi tidak dibuka hingga waktu hitung mundur selesai
	9. Sistem kembalike aplikasi pengunci ketika waktu habis
Kondisi Akhir	User Berhasil *Setting waktu*

TABEL II
SKENARIO USE CASE MEMILIH APLIKASI

Use case Login	
Aktor	Pengguna aplikasi
Tujuan	Menyetel waktu penguncian
Kondisi Awal	User memilih aplikasi yang akan di kunci
Skenario	
Aksi Aktor	**Reaksi Sistem**
1. Aktor Membuka aplikasi applock	
	2. Memeriksa aplikasi yang di install pada *smartphone.*
	3. Memeriksa aplikasi yang sudah dikunci
	4. menampilkan halaman aplikasi yang di install

5. memilih aplikasi yang ingin di kunci degan menekan *switch button*	
	6. Sistem menyimpan kondisi penguncian aplikasi yangdipilih
7. Jika aktor ingin mematikan penguncian tinggal menekan *switch* button	
	8. Sistem menyimpan kondisi membuka penguncian yangdipilih
Kondisi Akhir	User Berhasil *Setting waktu*

TABEL III
SKENARIO USE CASE MENGUNCI APLIKASI

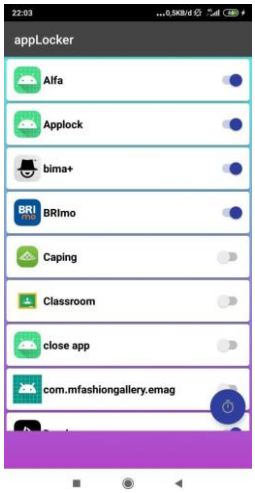
Use case Login	
Aktor	Pengguna aplikasi
Tujuan	Mengunci aplikasi yang telah di pilih dengan waktu yang telah di tentukan
Kondisi Awal	User sudah memilih aplikasi yang akan di kunci dan sudah menyetel lama waktu aplikasiyang akan di kunci
Skenario	
Aksi Aktor	**Reaksi Sistem**
1. Aktor Sudah memilihaplikasi dan waktu untuk penguncianaplikasi yang telah dipilih	
	2. Memeriksa aplikasi yang akan di kunci pada *handphone* .
	3. Memeriksa lama waktu penguncian aplikasi
4. Menjalankan waktu penguncian	
	5. Sistem menyimpan kondisi penguncian aplikasi yang dipilih
	6. Sistem mengunci aplikasi yang dipilih berdasarkan waktu yang telah di atur
Kondisi Akhir	User Berhasil mengunci aplikasi

4. Perancangan *Interface*

Perancangan *interface* merupakan gambaran *interface* tampilan dari *system* pengunci aplikasi yang akan di buat.

a. *Interface* Daftar Aplikasi

Gambar 5 merupakan *interface* daftar aplikasi yang di install pada *smartphone.*



Gambar 5. *Interface* Daftar Aplikasi

b. *Interface* Setting Waktu

Gambar 6 merupakan *Interface* pengaturan waktu yang akan dijalankan pada *system* pengunci. Ketika waktu

(menit) pengunci di atur maka menit akan otomatis di konversi menjadi satuan jam.

Gambar 6. *Interface* Setting Waktu

c. *Interface* Pengingat Aplikasi yang Terkunci
Gambar 7 merupakan *Interface* pengingat Aplikasi yang Terkunci.

Gambar 7. *Interface* Pengingat Aplikasi yang Terkunci

5. Implementasi

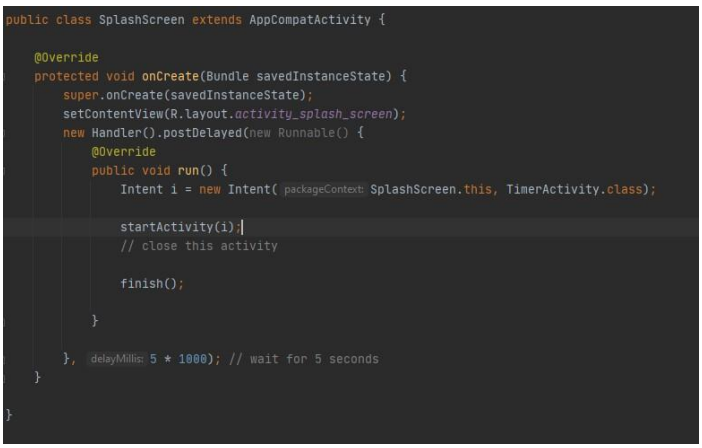
Implementasi merupakan aktivitas untuk menerjemahkan desain ke dalam Bahasa pemrograman. Berikut ditampilkan coding dalam *system* pengunci aplikasi pada *smartphone*. Gambar 8 sampai gambar 11 merupakan gambar sintak coding *system* pengunci aplikasi

```
public class appItem {
    private String mAppName;
    private String mPackageName;
    private Drawable mAppIcon;
    private boolean mLocked;

    //Constructor
    public appItem(String mAppName, String mPackageName, Drawable mAppIcon, boolean mLocked) {
        this.mAppName = mAppName;
        this.mPackageName = mPackageName;
        this.mAppIcon = mAppIcon;
        this.mLocked = mLocked;
    }

    //Getters and Setters
    public String getmAppName() { return mAppName; }

    public String getmPackageName() { return mPackageName; }

    public Drawable getmAppIcon() { return mAppIcon; }

    public boolean getmLocked() { return mLocked; }

    public void setmLocked(boolean mLocked) { this.mLocked = mLocked; }
}
```

Gambar 8. Coding inisialisasi daftar aplikasi

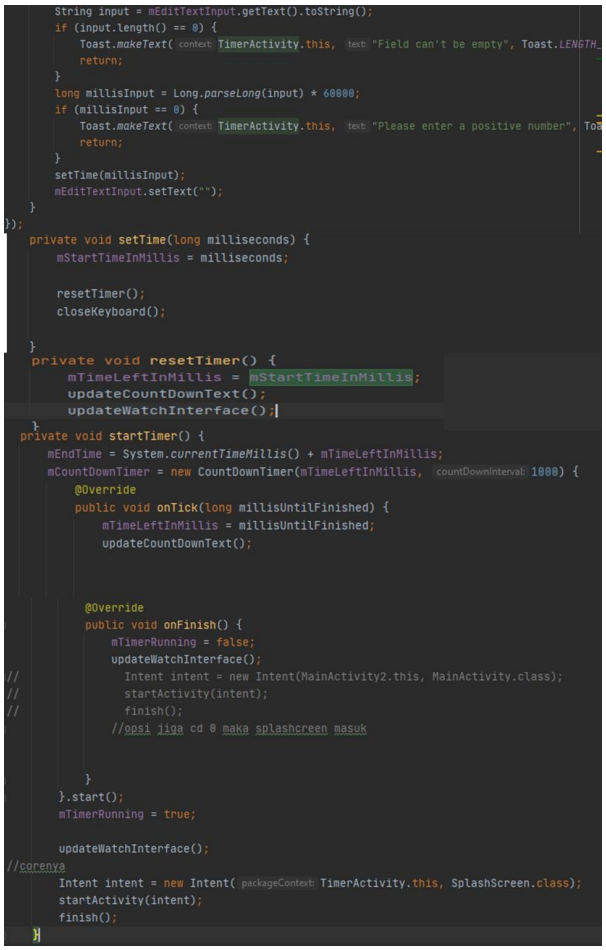
```
private ArrayList<appItem> appItemArrayList;
private appItems appItems;
private boolean lockFragmentState;

@Nullable
@Override
public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,
                         Bundle savedInstanceState) {

    return inflater.inflate(R.layout.activity_apps, container, attachToRoot: false);
}

@Override
public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    appItemArrayList = ((HomeActivity) Objects.requireNonNull(getActivity())).getAppItems();
    appItems = new appItems(getActivity());
    buildRecyclerView();
}

@Override
public void onActivityCreated(@Nullable Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
}

public void buildRecyclerView () {
    RecyclerView recyclerView = getView().findViewById(R.id.user_recycler_view);
    // use this setting to improve performance if you know that changes
    // in content do not change the layout size of the RecyclerView
    recyclerView.setHasFixedSize(true);
    // use a linear layout manager
    RecyclerView.LayoutManager recyclerViewLayoutManager = new LinearLayoutManager(getActivity())
    recyclerView.setLayoutManager(recyclerViewLayoutManager);

    // specify an adapter
    appAdapter adapter = new appAdapter(appItemArrayList);
    recyclerView.setAdapter(adapter);

    adapter.setOnItemClickListener(
            (position) -> appItems.launchAppIntent(appItemArrayList.get(position).getmPackageName
    );
    adapter.setOnCheckedChangeListener((CompoundButton buttonView,
                                        int position, boolean isChecked) -> {
        if (buttonView.isShown()) {
            if (isChecked) {
                appItems.addApp(appItemArrayList.get(position), position);
            }
            if(!isChecked){
                appItems.removeApp(appItemArrayList.get(position), position);
            }
            Toast.makeText(
                    getActivity(),
                    (isChecked)
                            ? "Locked " + appItemArrayList.get(position).getmAppName()
                            : "Unlocked " + appItemArrayList.get(position).getmAppName(),
                    Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
public void onSaveInstanceState(@NonNull Bundle outState) {
    super.onSaveInstanceState(outState);
}
}
```

Gambar 9. Coding service pengingat aplikasi yang dipilih



```
public class SplashScreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent i = new Intent( packageContext: SplashScreen.this, TimerActivity.class);

                startActivity(i);
                // close this activity

                finish();

            }

        }, delayMillis: 5 * 1000); // wait for 5 seconds
    }

}
```

Gambar 10. Coding kondisi pengalihan halaman



```
        String input = mEditTextInput.getText().toString();
        if (input.length() == 0) {
            Toast.makeText( context: TimerActivity.this,  text: "Field can't be empty", Toast.LENGTH_
            return;
        }
        long millisInput = Long.parseLong(input) * 60000;
        if (millisInput == 0) {
            Toast.makeText( context: TimerActivity.this,  text: "Please enter a positive number", Toa
            return;
        }
        setTime(millisInput);
        mEditTextInput.setText("");
    }
});
    private void setTime(long milliseconds) {
        mStartTimeInMillis = milliseconds;

        resetTimer();
        closeKeyboard();

    }
    private void resetTimer() {
        mTimeLeftInMillis = mStartTimeInMillis;
        updateCountDownText();
        updateWatchInterface();
    }
    private void startTimer() {
        mEndTime = System.currentTimeMillis() + mTimeLeftInMillis;
        mCountDownTimer = new CountDownTimer(mTimeLeftInMillis,  countDownInterval: 1000) {
            @Override
            public void onTick(long millisUntilFinished) {
                mTimeLeftInMillis = millisUntilFinished;
                updateCountDownText();



            @Override
            public void onFinish() {
                mTimerRunning = false;
                updateWatchInterface();
//                Intent intent = new Intent(MainActivity2.this, MainActivity.class);
//                startActivity(intent);
//                finish();
                //opsi jiga cd 0 maka splashcreen masuk


            }
        }.start();
        mTimerRunning = true;

        updateWatchInterface();
//corenya
        Intent intent = new Intent( packageContext: TimerActivity.this, SplashScreen.class);
        startActivity(intent);
        finish();
    }
```

Gambar 11. Coding *Timer*

IV. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan, dapat disampaikan hal-hal sebagai berikut:

a. Sistem pengunci aplikasi akan menampilkan daftar aplikasi yang telah di install kemudian memilih aplikasi yang akan di kunci selanjutnya akan diarahkan ke halaman isi waktu (*timer*) yang akan di atur dan tekan tombol start dan aplikasi yang dipilih. Selama durasi waktu (*timer*) yang telah di tetapkan, maka aplikasi yang dipilih untuk di kunci tidak akan dapat di buka (disembunyikan dari pandangan user) hingga waktu selesai.

b. Hitung waktu mundur yang telah di set berjalan sesuai waktu yang ditentukan menggunakan metode *timer* dan countdown di android studio.

REFERENSI

[1] Daeng, I. T. M., Mewengkang, N. N., & Kalesaran, E. R. (2017). Penggunaan *smartphone* dalam menunjang aktivitas perkuliahan oleh mahasiswa fispol unsrat manado. *Acta Diurna Komunikasi*, *6*(1).
[2] Bahagia, B., Satria, D., & Ahmadian, H. (2017). Perancangan sistem informasi manajemen data korban bencana berbasis mobile Android. *JEMSI (Jurnal Ekonomi, Manajemen, dan Akuntansi)*, *3*(2), 22-30
[3] https://salamadian.com
[4] https://developer.android.com.
[5] Kendall, K. E. dan Kendall, J. E. Analisis dan Perancangan Sistem. PT Indeks, Jakarta, 2010
[6] https://www.dicoding.com
[7] Syarif, M., & Nugraha, W. (2020). Pemodelan Diagram Uml Sistem Pembayaran Tunai Pada Transaksi E-Commerce. *JTIK (Jurnal Teknik Informatika Kaputama)*, *4*(1), 64-70.