BAB III
METODE PENELITIAN

3.1. Metode Penelitian

Metode penelitian pada penelitian ini adalah metode penelitian kuantitatif dengan cara studi literatur, penyebaran kuesioner, dan pengumpulan data dari uji kebergunaan. Metode studi literatur digunakan untuk melakukan analisis pada aspek *usability*, *deployability*, dan *usability* pada rancangan protokol sedangkan metode penyebaran kuesioner dan pengumpulan data uji kebergunaan digunakan untuk mendukung analisis pada aspek kebergunaan protokol.

3.2. Perancangan Protokol

Secara garis besar, protokol otentikasi *2fysh* terbagi menjadi dua bagian yaitu proses registrasi dan proses otentikasi.

3.2.1. Deskripsi Protokol

Protokol otentikasi yang dirancang pada penelitian ini dinamakan "***2fysh***", dibaca seperti "***two fish***" yang merupakan singkatan dari *"**2 factor you should have**"*. Berdasarkan namanya, terlihat bahwa fitur utama protokol ini adalah dua buah faktor SYH (*something you have)* yang perlu dibawa. 2 faktor SYH tersebut adalah ponsel pintar dan sebuah kartu NFC. Ponsel pintar dengan aplikasi khusus *2fysh* akan menjadi perangkat pengotentikasi. Namun, pada protokol *2fysh* ini, aplikasi pengotentikasi hanya dapat mengotentikasi hanya jika mendapat input data khusus dari kartu NFC sehingga untuk pengimplementasian protokol ini, diperlukan fitur NFC pada ponsel pintar yang digunakan.

Kekuatan utama dari protokol ini adalah dua buah faktor SYH. Faktor SYH pertama adalah ponsel dan yang kedua adalah kartu NFC. Sehingga walaupun satu faktor hilang atau dicuri, masih terdapat satu faktor SYH lagi sehingga pencuri tidak bisa mengakses ke dalam sistem.

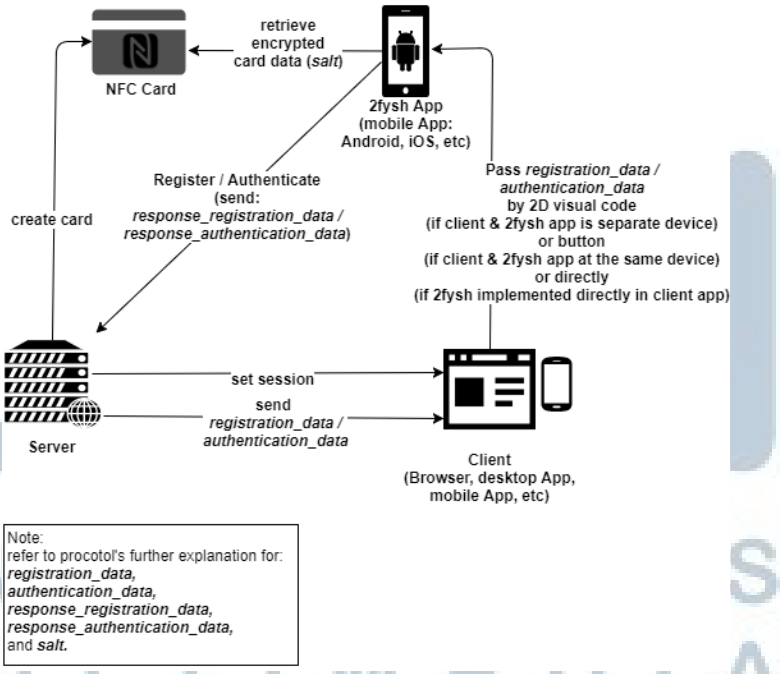
Selain itu, selain 2 buah faktor SYH, penggunaan ponsel sendiri biasanya dilengkapi dengan fitur otentikasi lokal ponsel seperti PIN, sidik jari, kata sandi, *touch gesture*, dan sebagainya. Fitur-fitur tersebut merupakan faktor yang lain selain faktor SYH yaitu faktor SYK (seperti PIN, *touch gesture*) atau SYA(seperti sidik jari, iris). Faktor SYK ini selalu digunakan

setiap hari oleh pengguna ponsel sehingga tidak meninggalkan beban yang lebih untuk mengingat jika faktor tersebut adalah faktor SYK. Maka dari pada itu, sangat dianjurkan jika pada penggunaan protokol *2fysh* ini digunakan pula faktor otentikasi bawaan dari ponsel demi mencapai tingkat keamanan yang lebih baik jika pada kasus terburuknya ponsel dicuri.

Untuk mengatasi kemungkinan buruk dari faktor SYH kedua yaitu kartu, jika dicuri ataupun hilang, data di dalam kartu terenkripsi dengan *public key* yang hanya dapat didekripsi oleh ponsel yang terkait dengan *private key* yang tersimpan secara aman di *Android KeyStore*.

Protokol otentikasi *2fysh* sendiri juga menggunakan metode otentikasi yang dirancang dengan memodifikasi protokol otentikasi FIDO *U2F* yang dirancang oleh FIDO Alliance yang sudah teruji dan aman dan menggabungkannya dengan keunggulan dari protokol otentikasi pico yang menggunakan dua buah faktor SYH.

Berikut adalah topologi yang menunjukkan hubungan setiap entitas yang diperlukan pada protokol otentikasi *2fysh*:



Gambar 3.1 Topologi protokol otentikasi *2fysh*

Pada protokol otentikasi *2fysh*, terdapat 4 entitas utama yaitu server, client, aplikasi Android *2fysh*, dan kartu NFC. Server berperan sebagai sistem yang akan diproteksi dengan protokol otentikasi *2fysh*. Server bertugas

mengirimkan data yang diperlukan untuk registrasi dan otentikasi. Data tersebut dikomunikasikan ke client melalui tiga cara untuk menyanggupi kebutuhan yang berbeda yaitu:

1. Melalui kode visual 2 dimensi

   Cara ini dilakukan ketika perangkat client yang melakukan otentikasi berbeda dengan perangkat aplikasi *2fysh* berada.

   Contoh: perangkat yang ingin melakukan otentikasi adalah web browser pada laptop, dan aplikasi *2fysh* berada pada ponsel. Sehingga untuk menyampaikan data dari laptop ke ponsel dilakukan dengan cara pembacaan kode visual 2 dimensi.

2. Melalui button / link URI (*Universal Resource Identifier*)

   Cara ini dilakukan ketika perangkat client yang melakukan otentikasi sama dengan perangkat aplikasi *2fysh* berada **tetapi** aplikasi client yang ingin melakukan otentikasi tidak mengimplementasikan protokol *2fysh* secara *native* pada aplikasi client.

   Contoh: perangkat yang ingin melakukan otentikasi adalah web browser pada ponsel dan aplikasi *2fysh* berada pada ponsel. Sehingga untuk menyampaikan data dari web browser ponsel ke aplikasi *2fysh* di ponsel, dilakukan dengan cara klik link URI.

3. Secara langsung dari aplikasi

   Cara ini dilakukan ketika perangkat client berupa ponsel dan aplikasi client menerapkan protokol *2fysh* secara langsung pada aplikasi. Dengan cara ini, aplikasi dapat langsung memroses data yang dibutuhkan.

   Contoh: aplikasi pada ponsel langsung menerapkan protokol *2fysh*. Sehingga data langsung tersampaikan pada aplikasi *2fysh* secara *native* sehingga tidak perlu lagi disampaikan ke aplikasi pengotentikasi *2fysh* karena data yang dibutuhkan sudah ada pada *environment* yang bersangkutan.

Ada pun data yang dirancang untuk dikomunikasikan agar protokol *2fysh* dapat bekerja dengan baik dan aman dari serangan. Data pada proses registrasi dari **server ke client** yang dinamakan **registration_data** membutuhkan data beserta fungsi sebagai berikut:

1. Action

Kolom action ini hanya dapat memiliki dua jenis data yaitu antara "*registration"* atau "*authentication*". Pada kolom ini untuk **registration_data**, kolom ini harus berisikan "registration"

2. Username

   Kolom ini berisikan username pengguna. Username sebenarnya tidak terlalu dibutuhkan pada protokol dan boleh dihilangkan. Namun, kolom ini dibutuhkan untuk keperluan antar muka ke pengguna. Digunakan sebagai *primary key* pengguna pada database saat pencarian maupun otentikasi

3. AppId

   Kolom ini berisikan App. Jika pada browser maka Appnya merupakan nama domain utama server tersebut, contohnya: "https://sunderi.com".

4. *Challenge*

   Challenge digenerate oleh server dan berperan sebagai problem yang harus dijawab oleh aplikasi pengotentikasi. Server akan memberikan *challenge* dalam bentuk *plain text* dan pemecahan problem adalah untuk pembuktian kepemilikan *private key* dengan cara pemberian *signature* pada *plain text* tersebut. *Challenge* untuk berisikan random string yang digenerasi oleh *secure random byte generator* berukuran 128, 192, maupun 256 bit.

5. *Register_portal*

   Berisikan link portal ke mana aplikasi harus mengirimkan data registrasinya. Contoh: "https://sunderi.com/*2fysh*/sign-up.php".

Data pada proses registrasi dari **aplikasi *2fysh* ke server** dinamakan **registration_response_data**¸ dengan data **challenge** dan **keyHandle** akan terenkripsi dengan *private key* sebagai *digital signature*. **Registration_response_data** berisikan data beserta fungsinya sebagai berikut:

1. Username

   Penjelasan sama dengan pada **registration_data**

2. *Challenge*

   Penjelasan sama dengan pada **registration_data**

3. *Public key*

   Public key berisikan *public key* yang digenerate oleh aplikasi pengotentikasi *2fysh* menggunakan algoritma asymmetric key yang diinginkan, contoh: RSA, Diffie-Hellmann Key Exchange. *Public key* nantinya akan digunakan untuk memverifikasi *challenge* yang telah dibubuhi *signature private key* pada saat otentikasi dan untuk mengenkripsi data yang ada pada kartu. Panjang public key tergantung dari panjang key algoritma asymmetric key yang digunakan (contoh: RSA 1024 atau 2048 bit).

4. *KeyHandle*

   KeyHandle digenerasi oleh client. *KeyHandle* merupakan key yang dibuat agar memudahkan pencarian data yang diinginkan pada aplikasi, juga sebagai alias untuk memanggil key yang dimaksud pada client nantinya (pada *keystore*). *KeyHandle* akan di simpan pada server. *KeyHandle* digenerasi menggunakan *secure random byte generator* dan dapat berukuran 32, 64, dan 128 bit.

Selain data yang dikomunikasikan pada proses registrasi, ada pula data yang dibuat dan disimpan pada proses registrasi yaitu:

1. Counter

   Counter ini berfungsi untuk mendeteksi adanya kloning jika entah dengan cara bagaimana aplikasi dapat di-clone oleh penyerang. Saat proses registrasi selesai, baik server maupun client akan membuat sebuah variabel counter bernilai 0. Nilai counter di-*increment* setiap kali terjadi otentikasi pada sisi server dan sisi client. Pada saat proses otentikasi di server, nilai counter pada server dicek kesamaannya dengan di client. Jika di server lebih besar dari pada di client, maka besar kemungkinan token hp telah di-clone (asumsi implementasi sistem tidak terjadi kesalahan).

2. Salt

   Salt adalah random secret yang diakan digunakan nanti pada saat proses otentikasi. Salt digenerasi menggunakan *secure random byte generator* dan dapat berukuran 32 atau 64 bit mengingat kapasitas penyimpanan kartu yang terbatas karena nilai salt ini akan kemudian dienkripsi menggunakan *public key* yang diterima oleh server dan nilai

tersebut: **publickey(salt),** dituliskan ke dalam kartu yang di-*issue* oleh lembaga (*relying party).*

Ada pula data yang diperlukan untuk dikomunikasikan dari **server ke client** pada saat otentikasi yang dinamakan **authentication_data**. Data dan fungsinya adalah sebagai berikut:

1. Action

   Kolom action ini hanya dapat memiliki dua jenis data yaitu antara "*registration"* atau "*authentication*". Pada kolom ini untuk **authentication_data**, kolom ini harus berisikan "authentication".

2. Username

   Penjelasan sama dengan pada **registration_data.**

3. AppId

   Penjelasan sama dengan pada **registration_data.**

4. Challenge

   Penjelasan sama dengan pada **registration_data**.

5. KeyHandle

   Penjelasan sama dengan pada **registration_response_data.**

6. Authentication_portal

   Berisikan link portal ke mana aplikasi harus mengirimkan data otentikasinya. Contohnya: "https://sunderi.com/*2fysh*/sign-in.php".

Data pada proses registrasi dari **aplikasi *2fysh* ke server** dinamakan **authentication_response_data**, dengan data **challenge+salt** dan **counter** terenkripsi dengan *private key* sebagai *digital signature*. **Authentication_response_data** berisikan data beserta fungsinya sebagai berikut:

1. Username

   Penjelasan sama dengan pada **registration_data.**

2. Challenge+salt / signedstuff

   Untuk mengotentikasi, bit ***challenge*** yang didapatkan ditempeli dengan bit ***salt*** yang didapat dari hasil mendekripsi isi dari kartu NFC yang isinya berupa **publickey(salt)** dengan *private key*. Setelah itu challenge + salt tersebut dienkripsi dengan *private key* sebagai *digital signature*.

3. Counter

Bersamaan dengan challenge+salt, counter juga dienkripsi bersamaan dengan *private key*.

3.2.2. Registrasi

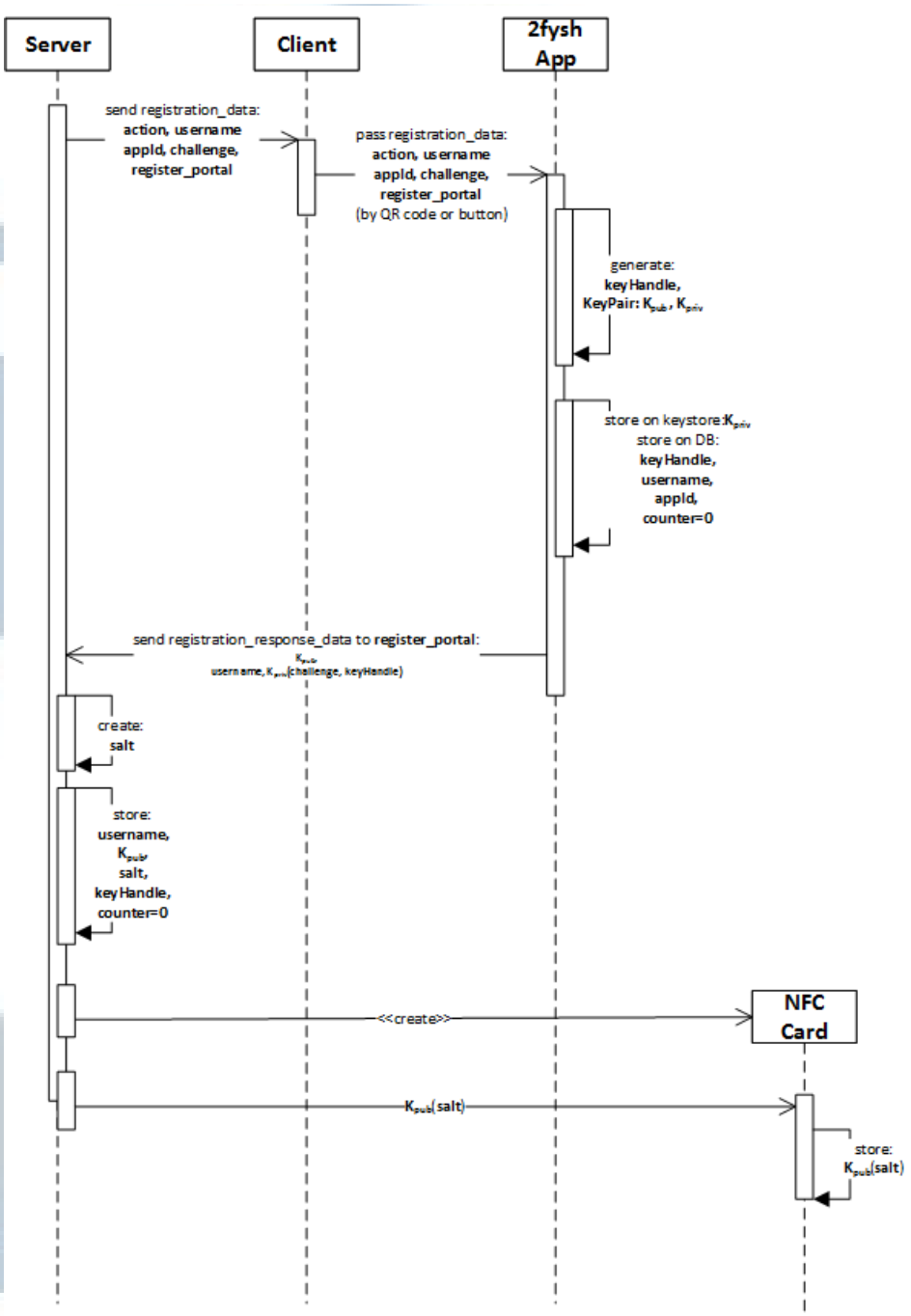
Pertama server akan mengirimkan **registration_data** yang berisikan data dalam format JSON yang berisikan **action, username, appId, challenge, register_portal.** Setelah itu data tersebut disalurkan ke aplikasi *2fysh* dengan salah satu dari tiga cara yang dijelaskan pada poin **3.2.1.** yaitu dengan kode visual 2D, link URI, atau secara lansgung pada aplikasi.

Setelah penyampaian data pada aplikasi, aplikasi akan membuat **keyHandle** dan setelah itu membuat *key pair* yang berisikan *public key* dan *private key* berdasarkan **keyHandle** yang dibuat tadi. Selain sebagai alias untuk keystore, **keyHandle** juga berfungsi sebagai pemudah sistem untuk mencari *credential* serta agar dapat menyimpan beberapa akun untuk satu situs yang sama. Aplikasi kemudian akan menyimpan data ke dalam basis data lokal dengan data sebagai berikut: keyHandle, username, appId, serta counter (dibuat dengan nilai awal 0).

Setelah proses tersebut, aplikasi mengirimkan **registration_response_data** ke portal **register_portal**, sebuah link yang didapatkan dari **registration_data**. Data **registration_ response_data** berisikan *public key*, **username**, **challenge**, dan keyHandle. Dengan data yang terenkripsi dengan *private key* (*digital signature*) adalah **challenge** dan **keyHandle**.

Server kemudian menerima, mengecek *signature* dan kemudian membuat **salt**, sebuah random string yang digenerasi menggunakan *secure random byte generator* dan akan digunakan pada proses otentikasi sehingga membuat protokol *2fysh* menjadi memiliki 2 buah faktor SYH. **Salt** ini akan dienkripsi dengan *public key*. Setelah itu **salt** yang terenkripsi ini akan ditulis ke dalam kartu oleh pihak yang bertanggung jawab mengeluarkan kartu seperti korporasi, bank, dan sebagainya.

Berikut adalah diagram sekuensial protokol *2fysh* untuk proses registrasi:



Gambar 3.2 Diagram sekuensial proses registrasi *2fysh*

3.2.3. Otentikasi

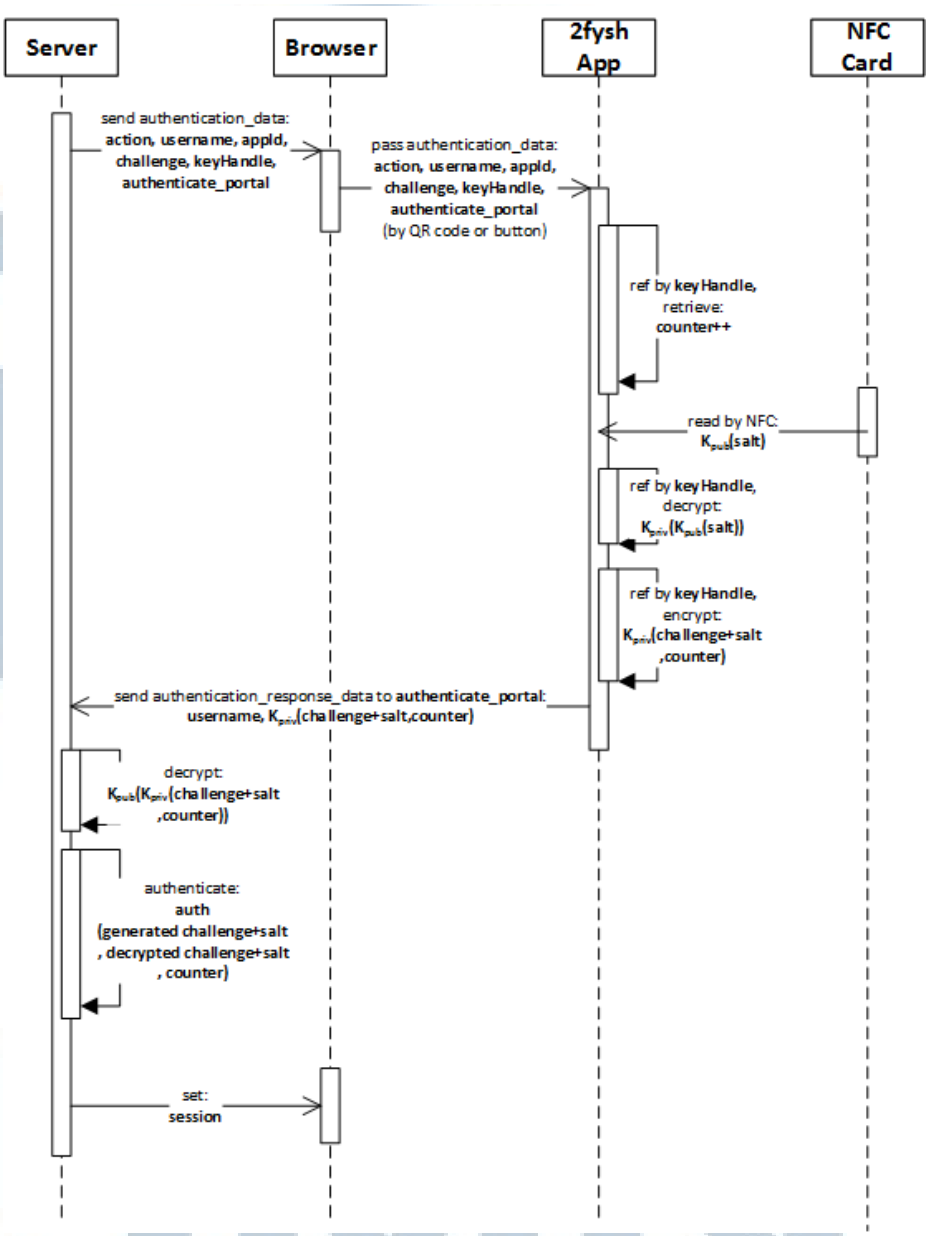
Pada proses otentikasi, server akan mengirimkan **authentication_data** yang berisikan **action, username, appId, challenge, keyHandle, dan authenticate_portal.** Setelah itu data tersebut disalurkan ke aplikasi *2fysh* dengan salah satu dari tiga cara yang dijelaskan pada poin **3.2.1.** yaitu dengan kode visual 2D, link URI, atau secara lansgung pada aplikasi.

Setelah itu, aplikasi akan mengambil **counter** berdasarkan **keyHandle** dari basis data lokal aplikasi. Aplikasi kemudian akan menunggu pembacaan NFC agar didapatkan isi dari faktor kedua SYH yang diperlukan yaitu *salt*. *Salt* diperoleh setelah didekripsi dengan *private key* melalui keystore menggunakan **keyHandle**.

Seluruh data yang dibutuhkan kemudian dienkripsi dengan *private key* melalui keystore menggunakan **keyHandle** sebagai *digital signature* dan kemudian dikirimkan ke portal otentikasi **authenticate_portal** seperti yang telah diperoleh sebelumnya. Setelah seluruh proses selesai, aplikasi akan menambah nilai **counter (counter++).**

Server kemudian akan menerima data dan memverifikasi *digital signature* yang digunakan sekaligus mengenkripsi data penting dengan *public key* yang dimiliki server. Jika data seperti challenge+salt dan counter sesuai dengan yang diinginkan server, maka otentikasi berhasil dan server mengeset *session* ke *browser client*.

Berikut adalah diagram sekuensial protokol *2fysh* untuk proses otentikasi:



Gambar 3.3 Diagram sekuensial proses otentikasi *2fysh*

3.3. Perancangan Sistem

Dengan menggunakan rancangan protokol di **bagian 3.2.**, pada penelitian ini akan dibuat *proof of concept* sistem otentikasi *2fysh* dengan server yang menggunakan skema web service; client yang berupa browser baik dari komputer desktop maupun mobile; serta aplikasi pengotentikasi *2fysh* berbasis Android.

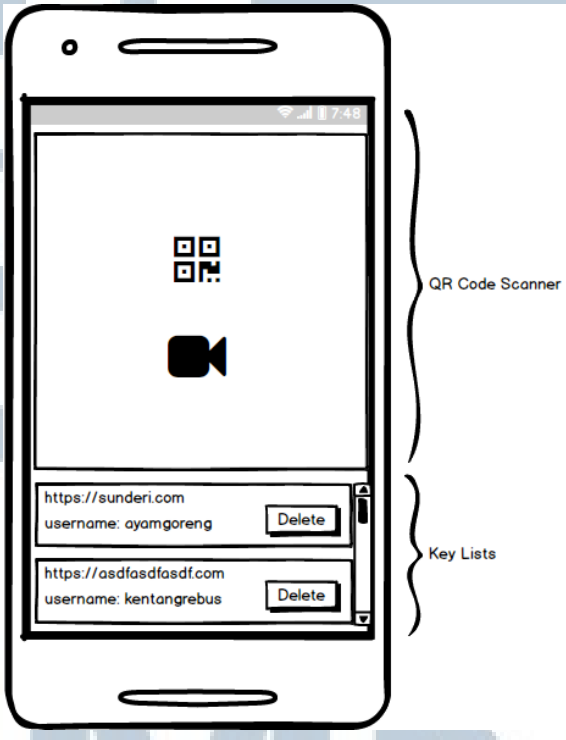
3.3.1. Perancangan Aplikasi pada Perangkat Android

Pada penelitian ini, dirancang purwarupa berupa aplikasi Android dengan nama ***2fysh* Authenticator**. Aplikasi ini berfungsi untuk melakukan

proses registrasi dan otentikasi. Aplikasi akan dapat melakukan proses registrasi dan menyimpan data kredensial pada basis data lokal yang dibutuhkan untuk melakukan otentikasi selanjutnya. Aplikasi dirancang pada OS Android dengan minimal API 18. Selain fitur tampilan *basic*, aplikasi menggunakan fitur kamera, pembacaan QR code, dan Android Key Store.

3.3.1.1 Perancangan Tampilan

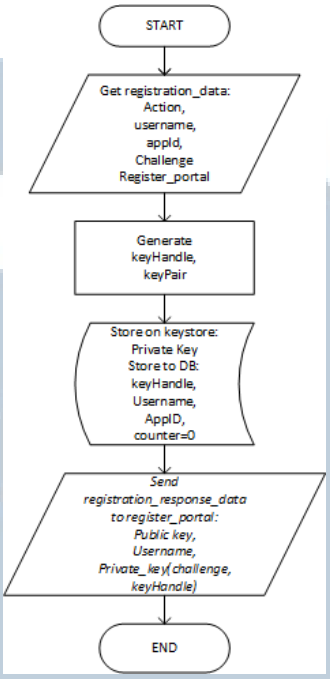
Ada pun perancangan tampilan (*wireframe)* aplikasi Android ***2fysh* Authenticator** sebagai berikut:



Gambar 3.4 Perancangan *wireframe* aplikasi *2fysh* Authenticator

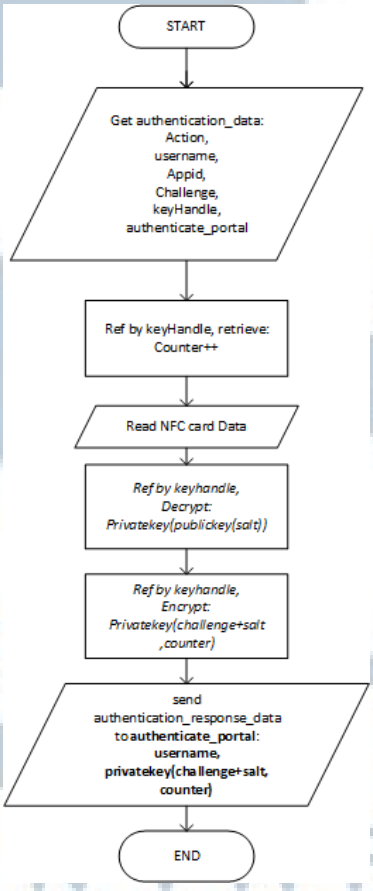
3.3.1.2 Diagram Alir

Diagram alir pada aplikasi saat melakukan registrasi adalah sebagai berikut:



Gambar 3.5 Diagram alir proses registrasi pada aplikasi

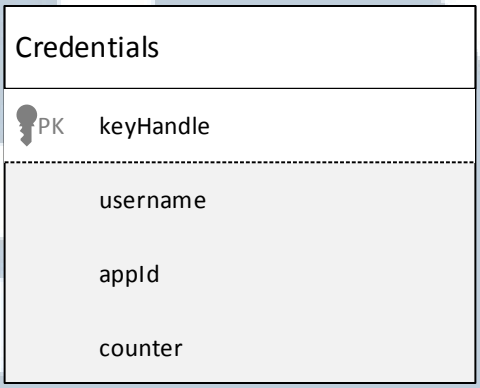
Diagram alir pada aplikasi saat melakukan otentikasi adalah sebagai berikut:



Gambar 3.6 Diagram alir proses otentikasi pada aplikasi

3.3.1.3 Perancangan Basis Data

Basis data yang diperlukan agar protokol dapat berjalan hanya satu tabel yaitu tabel "credentials". Ada pula ERD (*entitiy relationship diagram)* basis data yang dirancang sebagai berikut:



Gambar 3.7 ERD aplikasi *2fysh* Authenticator

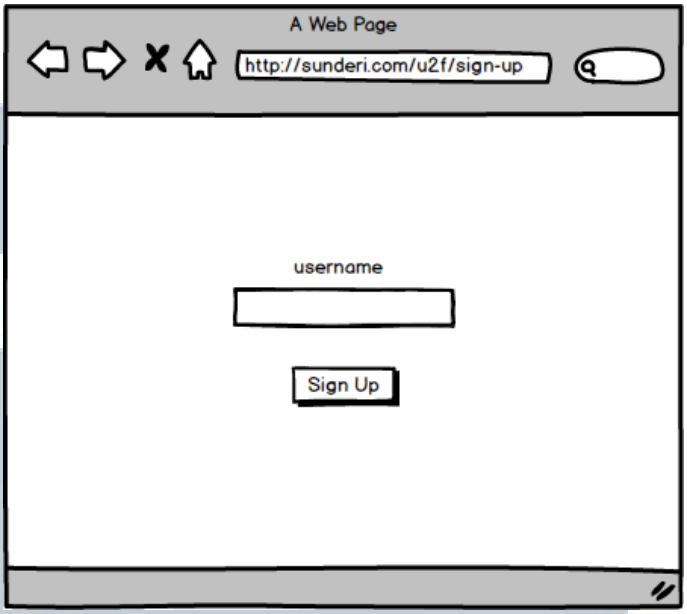
3.3.2. Perancangan Server Otentikasi

Pada penelitian ini, dirancang purwarupa server otentikasi *2fysh*. Server ini akan melakukan proses registrasi dan otentikasi bersama dengan aplikasi Android ***2fysh* Authenticator**. Server akan dapat memberikan data yang diperlukan untuk proses registrasi dan otentikasi, menerima dan menyimpan data hasil registrasi, serta mengotentikasi pengguna dan men-*set cookie*. Server akan dirancang dengan engine Apache, bahasa pemrograman PHP, dan library khusus yang dibutuhkan untuk membuat QR code.

3.3.2.1 Perancangan Tampilan

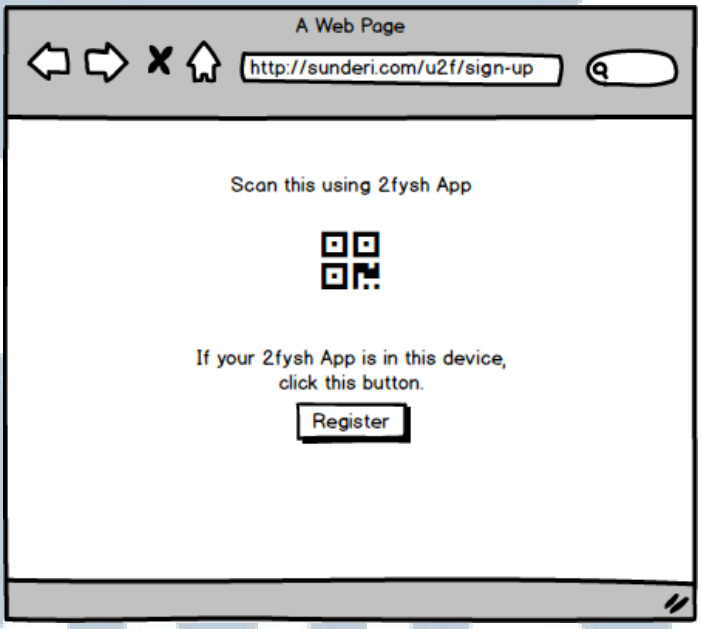
Ada pun perancangan tampilan sebanyak 4 halaman. 4 halaman tersebut penyajian tampilan dari server untuk pengguna sebagai berikut:

1. Pra-registrasi



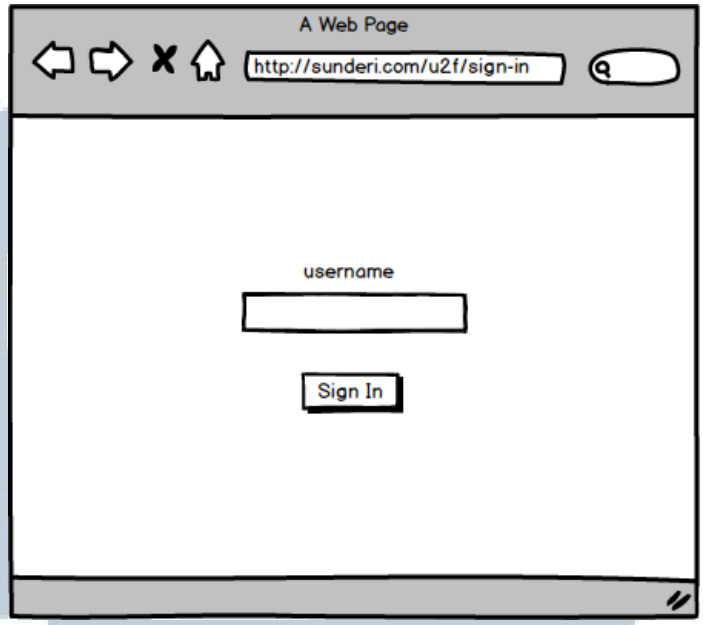
Gambar 3.8 Wireframe pra-registrasi tampilan web server

2. Registrasi



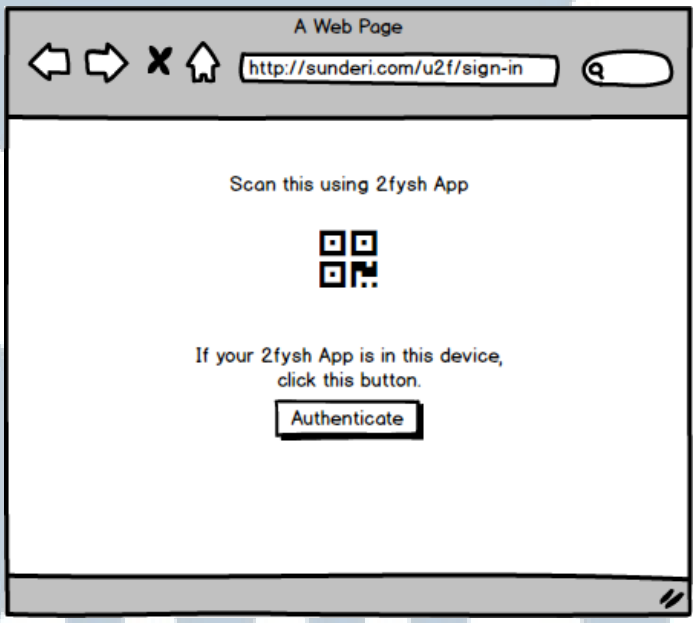
Gambar 3.9 Wireframe registrasi tampilan web server

3. Pra-otentikasi



Gambar 3.10 Wireframe pra-otentikasi tampilan web server

4. Otentikasi



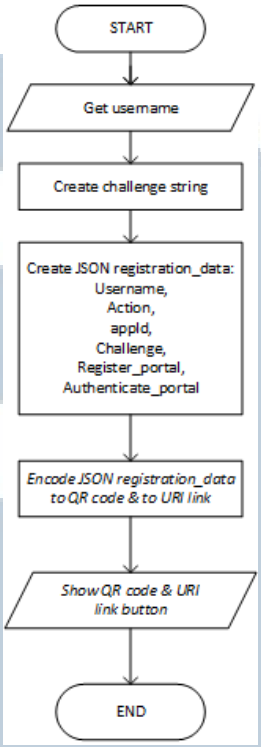
Gambar 3.11 Wireframe otentikasi tampilan web server

3.3.2.2 Diagram Alir

Ada pun 4 buah diagram alir yang diperlukan web server sebagai berikut:

1. Pra registrasi

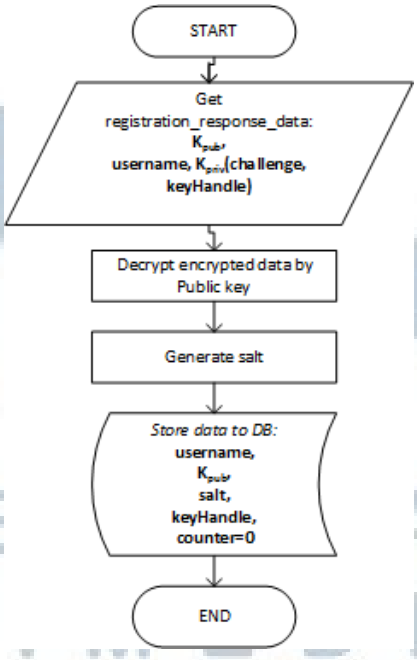
Berikut diagram alir dari proses pra registrasi server:



Gambar 3.12 Diagram alir pra registrasi server

2. Registrasi

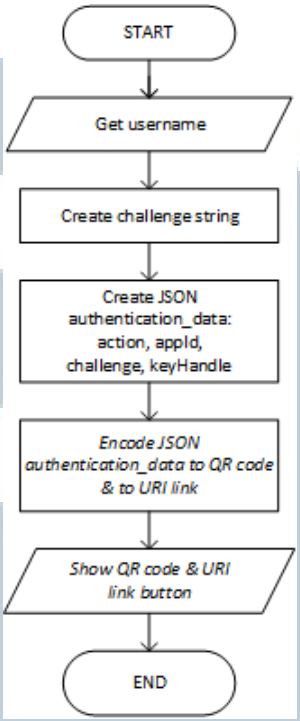
Berikut diagram alir dari proses registrasi server:



Gambar 3.13 Diagram Alir registrasi server

3. Pra otentikasi

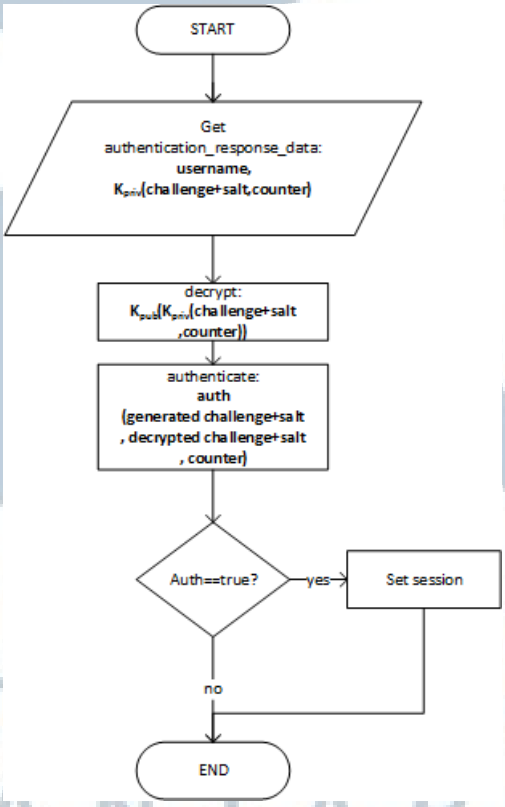
Berikut diagram alir dari proses pra otentikasi server:



Gambar 3.14 Diagram alir pra otentikasi server

4. Otentikasi

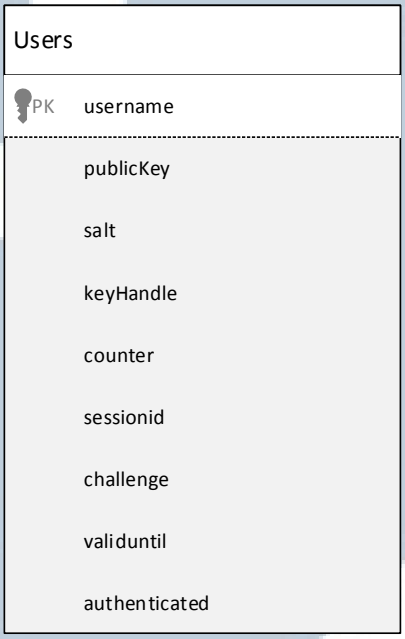
Berikut diagram alir dari proses otentikasi server:



Gambar 3.15 Diagram alir otentikasi server

3.3.2.3 Perancangan Basis Data

Basis data yang diperlukan agar protokol dapat berjalan hanya satu tabel yaitu tabel "credentials". Ada pula ERD (*entitiy relationship diagram)* basis data yang hanya berisikan satu buah tabel, dirancang sebagai berikut:



Gambar 3.16 ERD server *2fysh*

3.4. Syarat Pengimplementasian

Pengimplementasian protokol *2fysh* harus dilakukan dengan jalur komunikasi yang terenkripsi seperti TLS/SSL. Jika tidak diimplementasikan pada jalur terenkripsi. Ada skenario penyerangan terhadap sistem otentikasi protokol 2fysh sehingga penyerang dapat mendapatkan isi salt kartu sehingga dapat menjadikan sistem 2fysh sistem otentikasi satu faktor.

Skema penyerangan jika tidak menggunakan jalur komunikasi terenkripsi adalah sebagai berikut:

1. Penyerang menyadap seluruh komunikasi data dari saat tahap registrasi.
2. Ketika *client* mengirimkan *registration_response_data* ke server, penyerang mendapatkan public key.
3. Ketika *client* mengirimkan *authentication_response_data* saat melakukan otentikasi ke server, penyerang mendapatkan *signedstuff*.
4. Penyerang mendekripsi *signedstuff* dengan public key yang telah didapatkan sehingga penyerang mendapatkan nilai challenge+salt dan keyhandle.
5. Penyerang kemudian mengekstraksi nilai salt dari nilai challenge+salt.

6. Selanjutnya salt tersebut dienkripsi kembali dengan public key dan ditulis ke dalam kartu.

3.5. Aspek Analisis Protokol

Penelitian ini akan diuji dengan menganalisis aspek keamanan, kebergunaan, dan kemudahan untuk diterapkan (*security*, *usability*, dan *deployability)*. Ketiga aspek ini akan dianalisis dengan metode studi literatur. Selain itu, analisis aspek *usability* akan dilengkapi dengan uji purwarupa secara langsung ke subjek tes.

3.6. Pengumpulan Data Penelitian

Data penelitian akan dikumpulkan dengan cara pelaksanaan uji kebergunaan secara langsung kepada pengguna komputer sebanyak 10 orang. Kriteria peserta uji adalah mahasiswa dan karyawan kantor yang biasa menggunakan kartu sebagai metode otentikasi untuk akses masuk suatu sistem atau absensi. **Mahasiswa dan karyawan kantor sekaligus menjadi target utama dari sistem *2fysh*** dan adalah target pengguna dicobakan pada penelitian kali ini. Dari 10 orang peserta uji, dibutuhkan 5 orang yang fasih komputer sampai tahap sebagai berikut:

1. Dapat mengoperasikan sistem operasi Windows dan Android dengan lancar
2. Dapat menggunakan komputer dengan tujuan minimal untuk mengakses media sosial, *browsing*, kirim mengirim email, menggunakan aplikasi *word processor, spreadsheet*, dan presentasi.

Selain itu, 5 orang sisanya dibutuhkan orang yang fasih komputer sampai dengan tahap sebagai berikut:

1. Dapat membuat program atau aplikasi;
2. Dapat menginstall OS.

Pengumpulan data penelitan ditujukan untuk mengetahui aspek kebergunaan sebagai berikut:

1. *Effort* yang dibutuhkan untuk melakukan otentikasi *2fysh;*
2. Kemudahan untuk mempelajari sistem otentikasi *2fysh;*
3. Efisiensi sistem otentikasi *2fysh* yang akan diukur dengan waktu melakukan otentikasi;
4. Error yang terjadi ketika melakukan otentikasi *2fysh*;
5. Serta, perbandingannya dengan sistem otentikasi *u2f*.

3.7. Instrumen Penelitian

Penelitian dilakukan menggunakan instrumen perangkat keras maupun perangkat lunak berikut :

1. Satu buah PC berspesifikasi prosesor Intel core i5, RAM 8GB
2. Satu buah ponsel Android dengan minimum API 18 dengan kapabilitas NFC
3. Dua buah kartu / tag NFC NTAG 215
4. Text editor (notepad++, VS Code)
5. XAMPP (PHP dan MySQL)
6. Yubikey NEO