BAB 1
PENDAHULUAN

1.1 Latar Belakang

GamingAnywhere adalah *cloud game platform* yang memungkinkan *user* untuk memainkan *game* dalam sebuah komputer dan *android* dengan spesifikasi dibawah *game* tersebut, akan tetapi *user* atau *client* tidak dapat langsung terhubung ke *game server* dikarenakan *gaminganywhere* tidak menyediakan *link* atau penghubung antara *game server* dan *client*. Untuk itu *user* atau *client* membutuhkan sebuah *user interface* untuk menghubungkan *game server* dan *user* agar dapat saling tekoneksi dalam satu jaringan.

Berbeda dengan *platform game* yang lain. *GamingAnywhere* bersifat *open* dan dapat digunakan untuk penelitian, *GamingAnywhere* juga bersifat *cross-platform* yang membuatnya dapat berjalan pada beberapa sistem operasi seperti *windows* dan *linux*. Namun *user* tidak dapat langsung mengakses *game server* untuk langsung memainkan *game* tersebut, karena *game server* tidak menyediakan *user interface* untuk proses *login*. Untuk saat ini belum ada *user interface* untuk menghubungkan antar *game server* dan *user*.

Dengan permasalahan diatas, maka dibuatlah sebuah antar muka aplikasi *client android* untuk *mobile user* agar dapat *login* ke *server* dan memainkan *game* secara *mobile*. Fungsi dari *client android* adalah ketika *mobile user* telah memasang aplikasi *client android*, *mobile user* dapat langsung *login* ke *server*. Akhirnya *mobile user* terhubung ke *server* dan dapat langsung bermain *game.*

Client android ini merupakan aplikasi yang menghubungkan antara *mobile user* dan *server. Client android* ini dibuat dengan menggunakan konfigurasi *android* yang telah tersedia di *GamingAnywhere*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya ada pertanyaan terkait proyek akhir ini, yaitu bagaimana menyediakan sebuah *user interface* yang dapat menghubungkan *mobile user* ke *game server* dan mengakses *game* pada perangkat *mobile* ?

1.3 Tujuan

Adapun tujuan dari proyek akhir ini adalah membangun *user interface client android* untuk *platform cloud GamingAnywhere* yang dapat menghubungkan *mobile user* ke *game server* sehingga *user* dapat mengakses *game* pada perangkat *mobile* dan dapat langsung memainkan *game* secara *mobile.*

1.4 Batasan Masalah

Agar sistem dan pengerjaan proyek ini dapat dikerjakan sesuai dengan tujuan yang ingin dicapai maka diperlukan batasan masalah agar langkah pengerjaan menjadi sistematis. Adapun batasan masalah yang digunakan dalam proyek akhir ini adalah :

1. Tidak membahas pembuatan aplikasi *game android* ,
2. Tidak membahas keamanan aplikasi,
3. Fitur aplikasi yang tersedia adalah daftar, *login*, profile user, dan *last played,*
4. Versi android yang digunakan adalah android versi 4.1 , dan
5. *Game* yang digunakan yaitu *single player.*

1.5 Definisi Operasional

Client android berupa aplikasi yang memiliki fitur daftar, *login*, dan fitur pilih *game*. *Client android* merupakan perantara yang menghubungkan *mobile user* dan *game server* yang kemudian meneruskan proses *login user* ke *server*, setelah itu *user* dapat langsung memainkan *game* yang tersedia di *server* melalui *mobile* dengan menggunakan aplikasi *client android*.

1.6 Metode Pengerjaan

Metode pengerjaan yang digunakan penulis dalam membangun layanan *game* berbasis *GamingAnywhere* adalah metode *classical life cycle* atau pada umumnya dikatakan dengan nama metode *waterfall*. Konsep dari metode *waterfall* sendiri adalah melihat suatu masalah secara sistematis dan terstruktur dari atas ke bawah.

Metode *waterfall* secara garis besar memiliki tahapan sebagai berikut : Analisa, *Desain*, *Code*, *Testing*, *Maintenance* (*Presman Roger S.1997*).

a. Analisa

Tahapan ini merupakan proses pengumpulan data untuk membangun aplikasi dengan cara studi *literature*.

b. Design

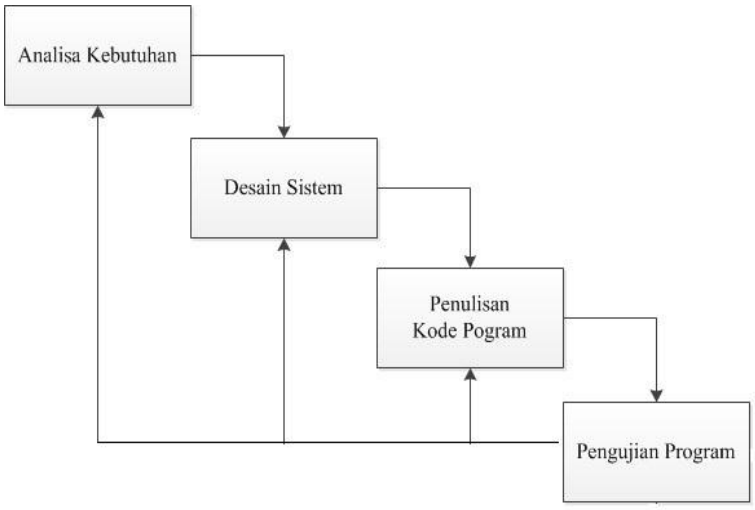
Tahapan ini merupakan pembuatan model dari aplikasi yang akan dibuat berdasarkan hasil dari analisa.

c. Coding

Tahapan ini merupakan implementasi pembuatan aplikasi dari *desain* yang telah dibuat sesuai dengan yang *user* butuhkan.

d. Testing

Tahapan *testing* dilakukan dengan menemukan kesalahan-kesalahan yang telah dibuat untuk kemudian dapat diperbaiki.



Gambar 1-1 Metode Waterfall

1.7 Jadwal Pengerjaan

Tabel 1-1 Jadwal pengerjaan

Kegiatan	2016-2017																			
	Januari				Februari				Maret				april				Mei			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengumpulan data																				
Perancangan desain																				
Coding																				
Testing																				
Dokumentasi																				