



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODE PENELITIAN

#### 3.1. Objek Penelitian

Objek penelitian pada pembuatan *games* Gomoku ini adalah pengguna *smartphone* Android yang berusia 17-22 tahun.

#### 3.2. Metode Perancangan Sistem

Terdapat beberapa model dalam metode pengembangan sistem SDLC, untuk memilih model yang sesuai dengan penelitian pertama-tama akan dilakukan perbandingan tiap model pada tabel 3.1 berikut:

Table 3.1 Tabel Perbandingan Model SDLC

Model SDLC	Kelebihan	Kekurangan
<i>Waterfall</i>	Memiliki proses yang urut, mulai dari analisa hingga <i>maintenance</i> . Setiap proses memiliki spesifikasinya sendiri, sehingga sebuah sistem dapat dikembangkan sesuai dengan apa yang dikehendaki (tepat sasaran). Lalu setiap proses tidak dapat saling tumpang tindih.	Proses yang dilakukan cenderung panjang dan juga lama, biaya penggunaan metode yang cenderung mahal, dan membutuhkan banyak riset dan juga penelitian pendukung untuk mengembangkan sistem.
<i>Spiral</i>	Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar, pengembang dan pemakai dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi	Memerlukan tenaga ahli untuk memperkirakan resiko dan harus mengandalkannya supaya sukses, merupakan model yang masih baru sehingga belum terbukti apakah

	karena perangkat lunak terus bekerja selama proses.	model ini efisien atau tidak.
<i>Iterative</i>	Beberapa fungsi yang bekerja dapat dikembangkan dengan cepat di awal siklus, hasil yang diperoleh lebih awal dan berkala, bisa merencanakan pengembangan secara paralel, kemajuan dapat diukur.	Membutuhkan <i>resource</i> yang cukup banyak, memerlukan perhatian manajemen, tidak cocok untuk proyek kecil.
<i>Rapid Prototyping</i>	Mengurangi waktu pengembangan produk, meminimalisasi perubahan-perubahan mendasar, meningkatkan efektifitas komunikasi di lingkungan industri atau dengan konsumen.	Memerlukan sumber daya yang cukup besar, resiko teknis yang tinggi, sistem yang tidak bisa dimodularisasi, memerlukan kerja keras dalam pengembangan.

Dari beberapa model pada tabel 3.1 di atas , pada penelitian ini diputuskan untuk menggunakan model *waterfall* yang dimodifikasi yaitu model *sashimi*, karena model *waterfall* akan dapat membuat sistem apa yang dikehendaki yaitu *games* Gomoku yang memiliki aturan dan juga fitur *multiplayer* yang diinginkan langsung jadi pada tahap akhir dari model ini. Penggunaan model *sashimi* juga dapat mengulang fase sebelumnya apabila terjadi kesalahan dan juga menghemat waktu. Tahap-tahap yang ada pada model *waterfall* adalah:

### 3.2.1. Perencanaan dan Analisis

Penelitian akan diawali dengan pengamatan-pengamatan langsung terhadap *video-video* mengenai cara kerja Firebase pada Youtube dengan nama *channel*

Firestore. Disana terdapat banyak pembicara yang membahas tentang Firestore lebih dalam dari berbagai hal seperti *tutorial* penggunaan fitur, apa saja fitur baru yang muncul pada versi Firestore terbaru, dan lain lain. Akan dilakukan juga pengamatan pada Playstore mengenai *game* Gomoku yang telah *publish* sebelumnya untuk mempelajari rancangan *design*, fitur, dan aturan permainan Gomoku. Hasil dari pengamatan tersebut langsung dicatat dalam penelitian dan juga dengan kegiatan pengamatan ini dapat diketahui ilmu atau cara kerja dari *game* Gomoku dan juga Firestore.

Selain melakukan kegiatan di atas penelitian ini juga melakukan studi kepustakaan melalui referensi-referensi yang ada di internet maupun di perpustakaan UMN. Referensi berupa aturan pada *games* Gomoku, jurnal penggunaan Firestore pada aplikasi lain, pembuatan *games* dengan menggunakan Android Studio, dan metode-metode pengembangan sistem.

Dalam pengembangan sistem *game* Gomoku ini juga diperlukan *tools* diantaranya:

1. *Backend services* yaitu Firestore.
2. Android Studio, Android Studio merupakan IDE (*Integrated Development Environment*) untuk membuat aplikasi android.
3. Java Development Kit (JDK), JDK berfungsi untuk mengkompilasi kode – kode Java menjadi aplikasi Java.
4. StarUML untuk pembuatan *design diagram*.

Pada tahap ini juga akan dilakukan analisis mengenai *user* siapa saja yang akan menggunakan aplikasi, fitur-fitur apa saja yang akan ada pada aplikasi, dan bagaimana *user interface* dari aplikasi *game* Gomoku ini sendiri.

### 3.2.2. Rancangan (*Design*)

Tahap *design* yaitu tahap dalam menentukan proses data yang diperlukan oleh sistem baru dengan tujuan memenuhi kebutuhan *user* dengan alat bantu UML dengan *software* visual StarUML. Pada penelitian ini digunakan pendekatan *Object Oriented* atau analisis berorientasi obyek dengan UML.

Pendekatan dilakukan dengan membuat beberapa *diagram* seperti Entity Relationship Diagram(ERD) yang merupakan gambaran *design database*, Flowchart yang merupakan gambaran arus cara kerja sistem, Use Case Diagram yang merupakan gambaran peran tiap *user* dalam sistem, Behavior Pattern yang merupakan gambaran proses yang bersifat rutinitas pada sistem, serta Table Criteria untuk pemilihan prioritas dari 12 kriteria dan Event Table untuk pembuatan Use Case Diagram.

### 3.2.3. Pengkodean (*Coding*)

Pada tahap ini akan dilakukan pengkodean dengan menggunakan aplikasi Android Studio dengan berdasarkan rancangan yang telah dibuat sedemikian rupa Tahap ini yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem yang berarti penggunaan komputer akan dimaksimalkan pada tahap ini. Pada penelitian akan dilakukan pemakaian Firebase juga sebagai *backend* dari *games* Gomoku yang akan dibuat.

### 3.2.4. *Testing dan Implementasi*

Tahap implementasi adalah tahap dimana rancangan sistem yang dibentuk menjadi suatu kode (*program*) yang siap untuk dioperasikan. Akan dilakukan proses *User Acceptance Test* (UAT) pada mahasiswa UMN yang menggunakan *smartphone* Android dan berumur 17-22 tahun.

Pengujian dilakukan dengan menggunakan *emulator* pada Android Studio sendiri atau dalam bentuk apk dan memberikan pada beberapa *user*. Dalam skripsi ini metode pengujian yang digunakan yaitu *black-box testing*.

### 3.2.5. *Pemeliharaan (Maintenance)*

Setelah melakukan implementasi terhadap sistem baru, tahap berikutnya yang perlu dilakukan adalah perbaikan dan perkembangan sistem. Pada tahap ini akan menambahkan fitur baru dan juga memperbaiki *bug* yang dialami *user* nantinya.

