BAB II
LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian ini akan menggunakan beberapa tinjauan studi yang akan digunakan untuk mendukung penelitian yang dilakukan. Tinjauan studi yang digunakan adalah sebagai berikut :

Penelitian Hanif Saifullah El Afrinuddin Zain (2013) yang membahas tentang Tingkat Pengetahuan Taktik Dan Strategi futsal Anggota Ekstra kurikuler futsal SMA Negeri 7 yogyakarta. Penerapan taktik dan strategi futsal anggota ekstrakurikuler SMA Negeri 7 Yogyakarta yang kurang sesuai saat pertandingan.

Penelitian ini bertujuan untuk mengetahui tingkat pengetahuan taktik dan strategi anggota ekstrakurikuler SMA Negeri 7 Yogyakarta dalam bermain futsal yang ditinjau dari 2 faktor yaitu jenis jenis taktik dan strategi menyerang dan bertahan dalam futsal.

Penelitian Muhammad Rustam Hatala (2011) yang membahas tentang video interaktif pembelajaran teknik dasar futsal. Multimedia yang dimanfaatkan sebagai media pembelajaran memiliki cakupan yang sangat luas sedangkan pada olahraga futsal juga memiliki banyak sekali teknik dalam bermain futsal baik teknik individu maupun teknik secara tim. Video interaktif hanya terdiri dari video teknik dasar futsal, video teknik penjaga gawang futsal, video dril latihan futsal dan video peraturan dasar futsal.

Penelitian Ibnu Diki Pratama (2015) yang membahas tentang pengembangan multimedia pembelajaran berbasis adobe flash mata kuliah permainan sepakbola untuk mahasiswa PJKR FIK UNY. Menurut Taufika Yusuf (2008:3) pemakaian media pembelajaran memiliki berbagai tujuan kognitif, afektif, dan psikomotor. Selain itu menurut Agus S Suryobroto (2001: 15) media hendaknya dapat dimanipulasi, dapat dilihat, didengar, dan dibaca.

Kesimpulan dari penelitian di atas yang membedakan adalah belum ada panduan futsal yang menggunakan sistem operasi android.

2.2. Landasan Teori

2.2.1. Definisi Android

Menurut Hermawan (2011 : 1), Android merupakan OS *(Operating System) Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

2.2.2 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android.

2.2.3 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program yang memiliki aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu (Supriyanto, 2005:2). Adapun karakteristik aplikasi adalah sebagai berikut:

1) Aplikasi merupakan elemen sistem logik dan bukan elemen sistem fisik seperti perangkat keras.
2) Elemen aplikasi bisa rusak.
3) Elemen aplikasi bisa direkayasa atau dikembangkan dan bukan dibuat di pabrik, seperti halnya perangkat keras.
4) Aplikasi tidak bisa dirakit atau disusun.

Pada sistem android ini akan menampilkan pilihan menu utama sebagai berikut :

1) Sejarah tentang futsal
2) Teknik dasar futsal seperti Teknik *passing*, teknik *control*, teknik *dribbling*, teknik *chipping*, teknik *shooting*
3) Formasi futsal
4) Peraturan futsal seperti luas lapangan, ukuran bola, jumlah pemain pertim, pelengkapan bermain, lama bermail futsal
5) Video teknik *skill* professional.
6) Tentang informasi turnamen futsal di Indonesia.

2.2.4 Futsal

Menurut Dendy Sugono (2008: 401) "futsal adalah olahraga permainan sepakbola, dengan lapangan dan gawang lebih kecil, biasanya dimainkan di dalam ruangan besar, masing-masing tim terdiri atas lima orang". futsal adalah permainan yang sangat cepat dan dinamis. Dari segi lapangan yang relatif kecil, hampir tidak ada ruang untuk membuat kesalahan (Justinus Lhaksana , 2011: 7).

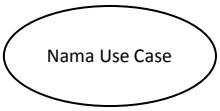
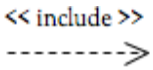
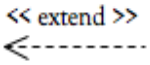
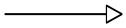
2.2.5 UML (*Unified Modeling Language)*

UML (*Unified Modeling Language*) adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek' (Nugroho, 2010). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks demikian rupa sehingga lebih mudah dipelajari dan dipahami.

2.2.5.1 *Use Case Diagram*

Use case diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. Simbol – simbol *Use Case Diagram* dapat dilihat pada Tabel 2.2

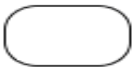
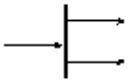
Tabel 2.2 Simbol – simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		*Use Case*	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2		*Actor*	Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
3		*Include*	Menspesifikasikan bahwa *use case* sumber secara eksplisit
4		*Extend*	Menspesifikasi bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
		*Generalisasi/ Generalization*	Hubungan generalisai dan spesialis antara dua buah *use case.*

b) Pengertian *Activity Diagram*

Activity Diagram adalah diagram yang menggambarkan aliran kerja atau aktifitas dari suatu sistem. Perlu diperhatikan bahwa *activity diagram* menggambarkan aktifitas sistem bukan apa yang dilakukan aktor. Simbol – simbol *Activity Diagram* dapat dilihat pada Tabel 2.3.

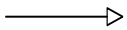
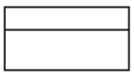
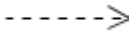
Tabel 2.3 Simbol – simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Initial Node*	Bagaimana objek dibentuk atau diawali.
3		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
4		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
5		*Decision node*	Suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi

2.2.5.2 *Class Diagram*

Class Diagram pada Tabel 2.4 menggambarkan struktur dan deskripsi *class*, *package* dan objek berserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem yang merupakan tipe diagram yang paling banyak dipakai. Simbol – simbol *Class Diagram* dapat dilihat pada Tabel 2.4.

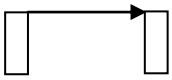
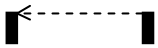
Tabel 2.4 Simbol – simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
3		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
4		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.2.5.3 *Sequence Diagram*

Sequence Diagram pada Tabel 2.5 mendeskripsikan bagaimana entitas dalam sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Diagram ini juga menunjukkan serangkaian pesan yang diperlukan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Simbol – simbol *Sequence Diagram* dapat dilihat pada Tabel 2.5.

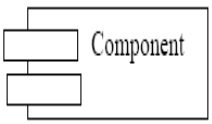
Tabel 2.5 Simbol – simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		*Actor*	Pengguna diluar sistem.
		*Return*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.2.5.4 *Component Diagram*

Component Diagram pada Tabel 2.6 Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Simbol – simbol *Component Diagram* dapat dilihat pada Tabel 2.6.

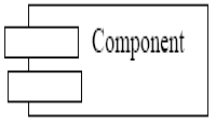
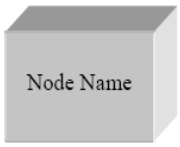
Tabel 2.6 Simbol – simbol *Component Diagram*

No	Nama	Gambar	Keterangan
1		*Component*	Sebuah *component* melambangkan sebuah entitas software dalam sebuah sistem.
2		*Depedency*	Sebuah *dependency* digunakan untuk menotasikan relasi antara dua *component*.

2.2.5.5 *Deployment Diagram*

Deployment Diagram pada Tabel 2.7 menggambarkan detail bagaimana komponen di-sebar (*di-deploy*) ke dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *node*, *server*, atau piranti kertas apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Simbol – simbol *Deployment Diagram* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol – simbol *Deployment Diagram*

No	Nama	Gambar	Keterangan
1		*Component*	Pada *deployment diagram*, *component* - *component* yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.
2		*Node*	*Node* menggambarkan bagian-bagian *hardware* dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
3		*Association*	Sebuah *association* digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen *hardware*.

2.2.6 Metode Pengujian *Five View*

Kitchenham B dan Pfleeger (2002:17) Pengujian *Five View* pengujian yang sifatnya deskriptif dimana *software* yang diuji dinilai melalui lima sudut pandang atau kategori yang berbeda yaitu, *User View, Manufacturing View, Transcendental View, Value-based View, Product View.*

1) *User View*

Kualitas menyangkut sejauh mana produk memenuhi kebutuhan dan harapan pengguna dan apakah suatu produk cocok untuk digunakan. Pendapat ini bersifat sangat *personal*. Hal ini berguna untuk mengidentifikasi fitur dari produk yang pengguna anggap penting. Pandangan ini dapat mencakup banyak unsur subjek, seperti kegunaan, keandalan, dan keefisiensasian

2) *Transcendental View*

Kualitas menurut pandangan ini adalah sesuatu yang dapat dikenali melalui pengalaman tapi tidak dapat selalu digambarkan. Objek atau *software* yang bagus itu menonjol dan dapat dengan mudah dikenali.

3) *Value-based View*

Value-base View merupakan penggabungan dari dua konsep yaitu keunggulan dan kelayakan. Kualitas adalah ukuran dari keunggulan dan nilai adalah ukuran layak. Beberapa banyak pengguna bersedia membayar untuk tingkat kualitas tertentu. Kualitas tidak berarti jika produk memenuhi nilai ekonomi. Pandangan bebabasis nilai antara biaya dan kualitas.

4) *Pruduct View*

Jika sebuah produk diproduksi dengan sifat internal (misalnya bahan dan tindakan) yang baik, maka produk akan memiliki sifat *eksternal* atau *output* yang baik dan dapat dieksplorasi hubungan antara sifat internal dan kualitas eksternal.

5) *Manufacturing View*

Pandangan ini berkaitan dengan faktor dalam industri, apakah produk memenuhi persyaratan atau tidak setiap penyimpangan dari persyaratan yang dinilai mengurangi kualitas produk. Konsep proses memainkan peran kunci produk yang dibuat harus orisinil sehingga biaya berkurang, misalnya biaya pembangunan dan biaya pemeliharaan. Kualitas produk dapat secara bertahap ditingkatkan dengan memperbaiki proses.