BAB II

TINJAUAN PUSTAKA

2.1 Rancang Bangun

Menurut Bambang (2013:27), rancang bangun adalah proses pembangunan system untuk menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian.

2.2 Firebase

Firebase adalah suatu layanan dari *Google* yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya *Firebase*, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua fitur yang menarik dari *Firebase* yaitu *Firebase Remote Config* dan *Firebase Realtime Database*. Selain itu terdapat fitur pendukung untuk aplikasi yang membutuhkan pemberitahuan yaitu *Firebase Notification*.

2.3 Aksara Bali

Dikenal sebagai *hanacaraka* dan menjadi salah satu huruf akasara tradisional yang berkembang di pulau bali, Indonesia. Aksara ini dipergunakan untuk menulis Bahasa bali dan Bahasa sansekerta. Huruf aksara bali yang biasa digunakan berjumlah 18 karakter yang dinamakan aksara wyanjana tetapi jika digunkaan dalam sehari-hari jumlahnya lebih dari 18 karakter. Dengan sedikit perubahan, aksara ini juga digunakan untuk menulis bahasa sasak yang berasal dari pulau Lombok. Aksara Bali berkerabat dekat dengan aksara Jawa.

2.3.1 Aksara *Wianjana* (Konsonan)

Aksara Bali dikenal juga sebagai *hanacaraka*, ditulis dan dibaca dari kiri kekanan.Terdapat 33 huruf konsonan dalam aksara Bali dengan 18 huruf dasar (disebut *wreṣāstra*) yang paling umum digunakan. Sisanya biasa dipakai dalam kata serapan tabel Sanskerta dan Kawi. Bentuk, nama, dan urutan huruf Konsonandikemukakan pada tabel 2.1 dibawah ini.

Tabel 2.1 Aksara Wianjana (Kosonan)

	= ha		= na		= ca		= ra		
	= ka		= da		= ta		= sa		
	= wa		= la		= ma		= ga		= ba
	= nga		= pa		= ja		= ya		= nya

2.3.2 Aksara Suara(Vokal)

Aksara suara disebut pula huruf vokal/huruf hidup dalam aksara Bali. Fungsi aksara suara sama seperti fungsi huruf vokal dalam huruf Latin. Jika suatu aksara wianjana (konsonan) diberi salah satu pangangge (tanda diakritik) aksara suara, maka cara baca aksara wianjana tersebut juga berubah, sesuai dengan fungsi pangangge yang melekati aksara wianjana tersebut. Berikut ini adalah aksara suara dalam aksara Bali.

Tabel 2.2 Aksara Suara (Vokal)

Warga Akṣara	***Aksara suara hresua* (huruf vokal pendek)**			**Nama**
	Akṣara Bali	**HurufLatin**	**Alfabet FonetisInternasional**	
Kantya (tenggorokan)	ᬅ	A	[a]	A kara
Talawya (langit-langit lembut)	ᬇ	I	[i]	I kara
Murdhanya (langit-langit keras)	ᬋ	Ṛ	[ɹ̩]	Ra repa
Dantya (gigi)	ᬍ	Ḷ	[l̩]	La lenga
Osthya (bibir)	ᬉ	U	[u]	U kara
Kanthya-talawya (tenggorokan & langit- langit lembut)	ᬏ	E	[e]; [ɛ]	E kara (E) Airsanya (Ai)
Kanthya-osthya (tenggorokan & bibir)	ᬑ	O	[o]; [ɔ]	O kara

2.3.3 Pangangge Suara

Bila suatu aksara wianjana (konsonan) dibubuhi pangangge aksara suara (vokal), maka cara baca aksara tersebut akan berubah. Contoh: huruf Na dibubuhi ulu dibaca Ni; Ka dibubuhi suku dibaca Ku; Ca dibubuhi

taling dibaca ce. Untuk huruf Ha ada pengecualian kadangkala bunyi /h/ diucapkan, kadangkala tidak. Hal itu tergantung kata dan kalimat yang ditulis.

Tabel 2.3 Pangangge Suara

	Ulu
	Ulu sari
	Ulu ricem
	Ulu Candra
	Pepet
	Pepet matedong
	Tedong
	Taleng
	Taleng matedong
	Taleng marepa
	Taleng marepa matedong
	Suku
	Suku ilut

2.4 Construct 2

Menurut (Arya, 2012: 29), Construct 2 adalah *software* pembuat *game* atau aplikasi berbasis HTML5 yang dikhususkan untuk platform 2D. *Software* ini dikembangkan oleh Scirra berbeda dengan Adobe Flash CS6, Construct tidak menggunakan bahasa pemrograman khusus, karena semua perintah yang digunakan pada game diatur oleh *Eventsheets* yang terdiri dari *event* dan *action*, *tools* ini digunakan dengan bantuan item yang tersedia, menambahkan perilaku mereka, dan membuat mereka menjadi bergerak dengan sebuah *event*.

2.5 Intel XDK

Menurut Sudjimat, D.A, dkk (2016:348) memberi batasan bahwa "intel XDK Adalah development kit yang dibuat oleh Intel untuk membuat aplikasi native untuk perangkat mobile menggunakan teknologi web seperti HTML5,CSS Dan JavaScript. Aplikasi web dikompilasi menggunakan platform Cordovadi server online untuk membuat aplikasi hybrid yang cross-platform".

Menurut Nuraeni, dkk (2016:196) memberi batasan bahwa "Intel XDK adalah sebuah development kit yang dibuat oleh Intel untuk membuat aplikasi native menggunakan teknologi web seperti HTML5, CSS, dan JavaScript".

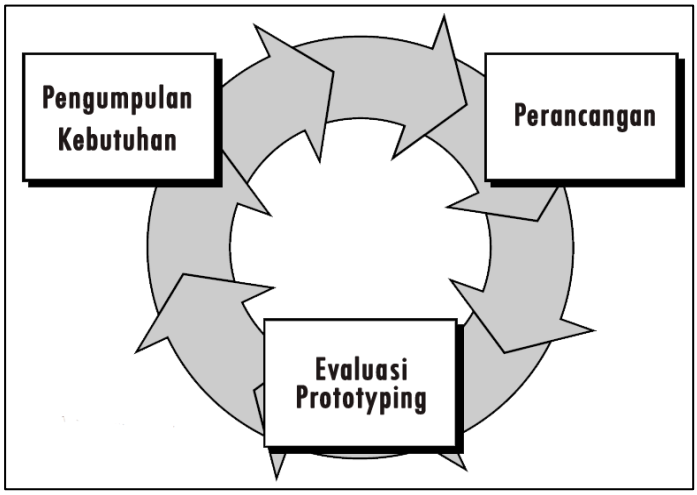
Dari beberapa pengertian intel XDK diatas, maka dapat disimpulkan bahwa intel XDK adalah sebuah delopment kit yang dibuat oleh intel untuk membuat aplikasi native yang menggunakan teknologi web.

2.6 Metode Pengembangan Sistem

2.6.1 Metode *Prototype*

(S,Rosa A. dan M.Shalahuddin 2013:31) Metode Prototype dimulai dari mengumpulkan kebutuhan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *prototype* agar lebih terbayang dengan apa yang sebenarnya diinginkan. Program Prototype biasanya merupakan program

yang belum jadi. program ini biasanya menyediakan tampilan dengan simulasi perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program dievaluasi oleh user,dapat dilihat pada gambar 2.1 berikut:



Gambar 2.1 Model *Prototype*

Tahap–tahap rekayasa *software* dalam *prototype model* pada gambar 2.1 di atas adalah sebagai berikut:

1. Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan. Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras.

2. Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan

3. Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi prototype yang dibuat dan digunakan untuk memperjelas kebutuhan software. Software yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan/kerusakan yang tidak ditemukan pada langkah sebelumnya.

2.7 UML (Unified Modelling Language)

UML ada karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya uml paling banyak digunakan pada metodologi berorintasi objek (Rosa dan Shalahuddin, 2019)

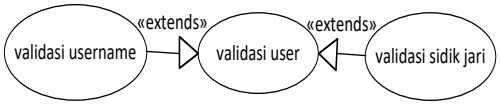
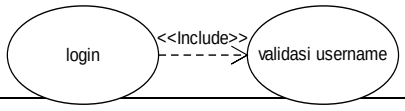
2.7.1 *Use Case* Diagram

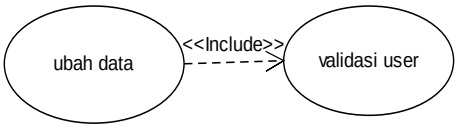
Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada use case adalah nama didefinisikan sesimpel

mungkin dan dapat dipahami (Rosa dan Shalahuddin, 2019). Adapun simbol-simbol sebagai berikut :

Tabel 2.4 Simbol *Use Case* Diagram

Keterangan	**Simbol**	**Deskripsi**
Use Case	e	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama *use case*
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi	————	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.

Ekstensi	«extends»	Relasi use case tambahan ke sebuah *use case*, dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal  Arah panah mengarah pada *use case* yang ditambahkan.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :  Arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).
Menggunakan/*include/uses*	<<Include>> «uses»	Ada dua sudut pandang yang cukup besar mengenai *include* di *use case* : a. *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut : 

		b. *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut :  Ke dua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi.

2.7.2 *Activity* Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2019). Simbol-simbol yang terdapat pada *activity diagram* adalah seperti pada tabel 2.5.

Tabel 2.5 Simbol *Activity* Diagram

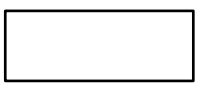
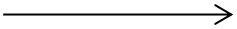
Keterangan	**Simbol**
Status awal	
Aktivitas	aktivitas

Percabangan	
Penggabungan	
Swimlane	Name Activities
Status akhir	

2.7.3 *Sequence* Diagram

Sequence diagram secara grafis menggambarkan bagaiman object berinteraksi dengan satu sama lain memalui pesan pada eksekusi sebuah *use case* atau proses. Adapun simbol–simbol dari *sequence* diagram terdapat pada tabel 2.7 berikut:

Tabel 2.6 Simbol *Sequence* Diagram

Nama Komponen	Komponen	Simbol
Lifeline	Menyatakan kehidupan suatu objek	
Activation	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung denngan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya	
Object	Menyatakan objek yang berinteraksi	
Pesan tipe create	Menyatakan suatu objek membuat objek yang lain	<< Create >> 

2.7.4 *Class* Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa dan Shalahuddin, 2019).

Tabel 2.7 Simbol *Class* Diagram

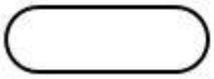
Nama Komponen	Komponen	Simbol
Class	Kelas pada struktur sistem	
Association	Relasi antar kelas dengan makna umum,asosiasi biasanya juga disertai multiplicity	
Directed Association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain,asosiasi biasanya juga disertai dengan multiplicity	

2.7.5 *State* Diagram

State diagram menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek sehingga setiap objek memiliki sebuah diagram status. Diagram Status mengambarkan seluruh state/status yang memungkinkan obyek-obyek dalam class dapat dimiliki dan kejadian-kejadian yang menyebabkan satus berubah. Perubahan dalam suatu state disebut juga transisi (transition). Suatu transisi juga dapat memiliki sebuah aksi yang dihubungkan pada status, lebih spesifik apa yang harus dilakukan dalam hubungannya dengan transisi status. Pada diagram ini, perilaku sistem

ditunjukkan. Sebuah status adalah kondisi selama hidup objek atau interaksi selama memenuhi suatu kondisi, melaksanakan suatu aksi, atau menunggu suatu kejadaian.

Tabel 2.8 Simbol *State* Diagram

Simbol	Nama	Keterangan
	State	digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisi saat itu.
	Titik awal (start)	Digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram flowchart.
	Titik Akhir(end)	Digunakan untuk menjelaskan/ menggambarkan akhir (end) dari kejadian dalam suatu diagram state chart.
[Guard]	Guard	Merupakan syarat transisi yang bersangkutan.
	Point	Digunakan untuk menggambarkan/ menjelaskan apakah akan masuk (entety point) ke dalam status atau keluar (exit point).
event	Event	Digunakan untuk menjelaskan kondisi yang menyebabkan sesuatu pada status.

2.8 Android

Menurut Safaat (2012) Android adalah sistem operasi menggunakan Linux yang dirancang untuk perangkat seluler seperti telepon pintar (smartphone) dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc.. dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Android ialah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah lisensi apache. Kode dengan lisensi terbuka dalam lisensi perizinan Android memungkinkan perangkat lunak ini untuk dimodifikasi secara bebas dan didistribusikan oleh pembuat perangkat, operator nirkabel dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat.

2.9 Jurnal Rancang Bangun Aplikasi Game Edukasi Pembelajaran Matematika Menggunakan Construct 2

(Jada Ario Yustin, Herry Sujaini, Muhammad Azhar Irwansyah.2016) Jurnal ini merupakan referensi dalam pembuatan skripsi ini. Didalam jurnal ini diterangkan bagaimana cara membangun media alternatif untuk pembelajaran matematika bagi anak-anak yang menyenangkan dan meningkatkan kemampuan anak dalam belajar matematika dasar.

2.10 Skripsi Rancang Bangun Media Pembelajaran Aksara Lampung Berbasis Mobile Android

(Febrian Eka Saputra.2017) Skripsi ini merupakan referensi dalam pembuatan skripsi ini. Didalam skripsi ini diterangkan bagaimana cara membangun media pembelajaran Aksara Lampung berbasis Mobile Android.