

## BAB 2

### LANDASAN TEORI

#### 2.1. Profil

Reog Ponorogo adalah salah satu bagian dari kesenian rakyat yang terdapat di Propinsi Jawa Timur, tepatnya di wilayah Propinsi Jawa Timur bagian Barat yaitu Kota Ponorogo. Kesenian Reog Ponorogo sampai saat ini dapat dijumpai pada acara adat maupun festival. Dilihat dari segi waktu dan model pertunjukan, reog mengalami banyak perubahan.

Reyog atau Reog berasal dari kata “riyet” atau dapat diartikan sebagai keadaan bangunan yang hampir rubuh, karena di dalam pertunjukan reog terdapat suara gamelan atau musik pengiring yang menyerupai “bata rubuh” yang artinya sangat semarak dan ramai (Poerwowijoyo, 1985:9). Akan tetapi sampai saat ini perdebatan antara pemilihan kosakata yang tepat mengenai reog masih berlangsung. Reog atau reyog keduanya sama-sama digunakan untuk menyebutkan salah satu jenis tarian yang berasal dari Ponorogo dengan ciri khas unsur topeng Dhadak Meraknya. Pada awalnya penduduk Ponorogo sering menyebutkan dengan istilah reog akan tetapi pada tahun 1995, saat Drs. Markoem Singodimejo menjabat sebagai Bupati Ponorogo beliau mengeluarkan SK Bupati yang menetapkan semboyan “REOG” yang memiliki arti resik, endah, ombor, girang-gemirang sebagai semboyan kota Ponorogo. Secara tidak langsung hal itu mempengaruhi masyarakat Ponorogo. Dahulu menggunakan istilah reog akan tetapi dengan adanya semboyan itu membuat masyarakat beralih pada kosakata reog

(Fauzannafi, 2005). Penulis memilih untuk menggunakan istilah yang lebih original yaitu reog. Sebagai kota dengan image Kota Reog, Ponorogo memiliki banyak sekali grup reog. Hampir setiap kecamatan memiliki grup, bahkan setiap desa memiliki lebih dari satu grup reog, seperti di wilayah Kecamatan Sumoroto. Kecamatan Sumoroto dianggap masyarakat Ponorogo sebagai tempat berdirinya pusat Kerajaan Bantar Angin, yaitu merupakan salah satu kerajaan yang dianggap sebagai tempat asal-usul reog Ponorogo (Fauzannafi, 2005).

## 2.2. Java

Java merupakan bahasa pemrograman yang dikembangkan agar mampu berjalan di atas berbagai platform perangkat keras dan perangkat lunak yang berbeda. Hal ini merupakan terobosan yang cukup besar, sebab aplikasi sebelumnya dikembangkan hanya untuk sistem operasi tertentu. Bahasa pemrograman Java saat ini bisa digunakan untuk mengembangkan aplikasi desktop, yaitu menggunakan JSE (Java Standard Edition) dan aplikasi yang berjalan di internet, yaitu menggunakan JEE (Java Enterprise Edition). Dalam mengembangkan aplikasi, Java menggunakan metodologi pengembangan berorientasi objek (OOP[Object Oriented Project]).

Java bisa digunakan pada berbagai keperluan. Pemrograman berbasis teks seperti pada Clipper atau FoxPro for DOS yang biasa ditemui di kasir-kasir minimarket bisa dilakukan dengan Java. Pemrograman berbasis GUI desktop seperti yang dikenal pada Visual Basic, Visual FoxPro, atau Delphi juga bisa dilakukan dengan Java. Pemrograman untuk peranti mikro seperti

ponsel atau tablet PC mudah dilakukan dengan Java. Tidak ketinggalan, pemrograman web bisa menggunakan Java Server Pages (JSP) atau Servlet. (Hudaya, 2015).

### **2.3. Android**

Android adalah sistem operasi mobile yang paling cepat tumbuh (OS). Dengan lebih dari 800.000 aplikasi yang tersedia di Google Play Store, ekosistem Android berkembang juga. Ada cukup keragaman dalam fitur perangkat dan operator nirkabel untuk menarik hanya tentang siapa pun. Netbooks selalu platform alami untuk mengadopsi Android, tetapi keaktifan di belakang Android telah makan pertumbuhan lebih lanjut ke tablet, televisi, dan bahkan mobil. Banyak perusahaan terbesar di dunia dari perbankan sampai makanan cepat saji sampai penerbangan telah menetapkan kehadiran di Android dan menawarkan layanan yang kompatibel. Pengembang Android memiliki banyak kesempatan, dan aplikasi yang relevan menjangkau lebih banyak orang daripada sebelumnya, meningkatkan kepuasan membuat aplikasi yang relevan (Schwarz dkk, 2013).

Android memiliki 4 komponen aplikasi. Setiap komponen memiliki peran yang berbeda-beda dan berdiri sendiri meskipun terkadang salah satu komponen bergantung pada komponen lainnya. Masing-masing komponen memiliki daur hidup masing-masing, yang mengatur kapan komponen tersebut dibuat dan bagaimana komponen tersebut dihapus. Untuk mengaktifkan komponen-komponen tersebut diperlukan sebuah pesan

asinkron yang disebut Intent (Gargenta, 2011). Berikut adalah 4 tipe komponen dalam aplikasi android:

1. Activity, merupakan suatu komponen yang merepresentasikan sebuah tampilan yang dilengkapi dengan user interface.
2. Service, merupakan suatu komponen yang berjalan di balik layar untuk menjalankan operasi yang menghabiskan banyak waktu.
3. Content Provider, merupakan suatu komponen yang mengatur sekumpulan data aplikasi.
4. Broadcast receiver, merupakan komponen yang merespon terhadap siaran (broadcast) pengumuman yang dikeluarkan oleh sistem.

Android mempunyai dukungan aplikasi dan game tidak berbayar terbanyak yang bisa di-download oleh penggunanya melalui Google Play Store. Dengan terdapatnya fitur seperti browser, MMS, SMS, GPS, dan lain-lain, maka sangat memudahkan penggunanya untuk mendapatkan informasi, mengetahui posisi, serta juga komunikasi antar para pengguna.

#### **2.4. Android SDK**

Menurut Safaat (2012), Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mengembangkan aplikasi pada platform Android yang menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di-release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat

bantu dan API untuk mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform Android aplikasi – netral, Android memberi Anda kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone/Smartphone.

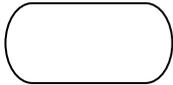
## 2.5. Android Studio

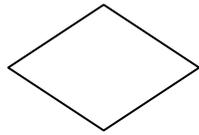
Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi android. (<https://developer.android.com/studio/intro?hl=ID>, 2018).

## 2.6. Flowchart

Ladjamudin (2013), Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma.

**Tabel 2.1.** Simbol *Flowchart*

Simbol <i>Flowchart</i>	Fungsi
<i>TERMINAL</i>	
	Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses/kegiatan.
<i>PREPARATION</i>	
	Simbol ini digunakan untuk mempersiapkan harga awal/nilai awal suatu variabel yang akan diproses.



*DECISION*

Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses.

---



*PROSES*

Simbol ini digunakan untuk menggambarkan suatu proses yang sedang dieksekusi.

---



*INPUT/OUTPUT*

Simbol ini digunakan untuk menggambarkan proses input (*read*) maupun proses output (*print*).

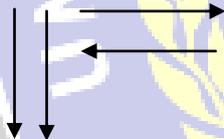
---



*SUBROUTINE*

Simbol ini digunakan untuk menggambarkan proses pemanggilan subprogram dari program.

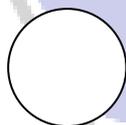
---



*FLOW LINE*

Simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.

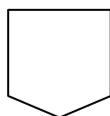
---



*CONNECTOR*

Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada di dalam suatu lembar halaman.

---



*PAGE CONNECTOR*

Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berpindah halaman.

---

	<b>MANUAL OPERATION</b>
	Simbol ini digunakan untuk menggambarkan suatu kegiatan atau proses bersifat manualisasi.
	<b>PRINTER</b>
	Digunakan untuk menggambarkan suatu kegiatan mencetak suatu informasi dengan mesin printer.
	<b>CONSOLE</b>
	Simbol ini digunakan untuk menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau CRT ( <i>Cathode Ray Tube</i> ).
	<b>DISK</b>
	Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media <i>magnetic disk</i> .
	<b>MANUAL INPUT</b>
	Simbol ini digunakan untuk menggambarkan proses pemasukan data melalui media keyboard.
	<b>TAPE</b>
	Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media <i>magnetic tape</i> .

## 2.7. Entity Relationship Diagram

Ladjamudin (2013), ERD (Entity Relationship Diagram) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Komponen - komponen Diagram Hubungan Entitas:

1. Entity

Pada E-R diagram, entity digambarkan dengan sebuah bentuk persegi panjang. Entity adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data.

2. Relationship

Pada E-R diagram, Relationship digambarkan dengan sebuah bentuk belah ketupat. Relationship adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (Relationship) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa digunakan kalimat aktif atau kalimat pasif)

3. Relationship Degree

Relationship Degree atau derajat relationship adalah jumlah entitas yang berpartisipasi dalam satu relationship.

4. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun Relationship. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun Relationship, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan Relationship.

5. Kardinalitas (Cardinality)

Kardinalitas Relasi menunjukkan jumlah maksimum tupelo yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat 3 (tiga) macam kardinalitas relasi, yaitu:

a. One to One

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

b. One to Many atau Many to One

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu.

c. Many to Many

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya.

## 2.8. *Data Flow Diagram (DFD)*

*Data Flow Diagram (DFD)* adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang

dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

*Data flow* diagram menjelaskan kepada *user* bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. *Data flow* diagram akan menginterpretasikan logical model dari suatu sistem. Arus dari data tersebut nantinya dapat dijelaskan dengan menggunakan kamus data (*data dictionary*). Simbol DFD:

**Tabel 2.2.** Simbol *Data Flow Diagram*

Notasi	Fungsi
	Simbol Entitas eksternal atau terminator menggambarkan asal atay tujuan data di luar sistem
	Simbol lingkaran menggambarkan entitas atau proses dimana aliran data masuk ditransformasikan ke aliran data keluar
	Simbol aliran data menggambarkan aliran data
	Simbol <i>file</i> menggambarkan tempat data disimpan

Sumber : (Sutabri, 2013)

## 2.9. Pengertian Kamus Data (Data Dictionary)

Sukanto dan Shalahuddin (2013), kamus data (data dictionary) di gunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar penulisan).

