

BAB II

LANDASAN TEORI

2.1 Rancang Bangun

(Pressman, 2002). Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen – komponen sistem diimplementasikan. (Ladjamudin, 2005) Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah – masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. (Pressman, 2002) Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian. (Whitten et al, 2004) Bangun sistem adalah membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain.

2.2 Media Pembelajaran

(Arsyad, 2014) mendefinisikan kata media berasal dari bahasa latin *medius* yang secara harfiah berarti ‘tengah’, perantara ‘atau’ pengantar. Media adalah perantara atau pengantar pesan dari pengantar atau pengirim ke penerima pesan menurut (Gerlach & Ely, 1971) mengatakan bahwa media apabila dipahami secara garis besar adalah manusia, materi, atau kejadian yang membangun kondisi kondisi yang membuat siswa mampu memperoleh pengetahuan, keterampilan atau sikap.

Pembelajaran adalah usaha-usaha yang terencana dalam memanipulasi sumber-sumber belajar agar terjadi proses belajar. Media pembelajaran dapat diartikan sebagai segala sesuatu yang dapat digunakan untuk menyalurkan pesan dari pengirim ke penerima sehingga merangsang pikiran, perasaan, perhatian, dan minat siswa sedemikian rupa sehingga

proses belajar terjadi. Manfaat media pembelajaran secara umum memiliki kegunaan-kegunaan sebagai berikut :

- a. Memperjelas penyajian materi pembelajaran agar tidak terlalu verbalistik (dalam bentuk kata-kata tertulis atau lisan).
- b. Media pembelajaran dapat digunakan secara tepat, bervariasi, dan lebih interaktif mampu mengatasi sikap pasif anak didik.

2.3 Aksara Lampung

(Abdullah, 2015) Aksara Lampung (Had Lampung) adalah bentuk tulisan yang memiliki hubungan dengan aksara Pallawa dari India Selatan. Macam tulisannya fonetik berjenis suku kata yang merupakan huruf hidup seperti dalam Huruf Arab, dengan menggunakan tanda-tanda fathah pada baris atas dan tanda-tanda kasrah pada baris bawah, tetapi tidak menggunakan tanda dammah pada baris depan, melainkan menggunakan tanda di belakang, di mana masing-masing tanda mempunyai nama tersendiri.

Had Lampung dipengaruhi dua unsur, yaitu Aksara Pallawa dan Huruf Arab. Had Lampung memiliki bentuk kekerabatan dengan aksara Rencong, Aksara Rejang Bengkulu, aksara Sunda, dan aksara Lontara. Had Lampung terdiri dari huruf induk, anak huruf, anak huruf ganda dan gugus konsonan, juga terdapat lambang, angka dan tanda baca. Had Lampung disebut dengan istilah Kaganga ditulis dan dibaca dari kiri ke kanan dengan Huruf Induk berjumlah 20 buah.

Aksara lampung telah mengalami perkembangan atau perubahan. Sebelumnya Had Lampung kuno jauh lebih kompleks, sehingga dilakukan penyempurnaan sampai yang dikenal sekarang. Huruf atau Had Lampung yang diajarkan di sekolah sekarang adalah hasil dari penyempurnaan tersebut.

2.3.1 Huruf Induk

Aksara Lampung Disebut dengan istilah kaganga, ditulis dan dibaca dari kiri kekanan (pada tabel 1 dibaca dari atas ke bawah). Huruf Induk berjumlah 20 buah. Bentuk, nama, dan urutan huruf induk dikemukakan pada tabel 2.1 dibawah ini.

Tabel 2.1 Huruf Induk

Aksara	Nama	Aksara	Nama
↗	Ka	↖	Ja
↘	Ga	↙	Nya
↗	Nga	ω	Ya
✓	Pa	↖	A
↘	Ba	↗	La
✓	Ma	↘	Ra
↗	Ta	↙	Sa
↘	Da	↗	Wa
↙	Na	↘	Ha
↖	Ca	↗	Gha

2.3.2 Anak Huruf

Anak huruf aksara lampung ada 12 buah yaitu seperti tabel 2.2 dibawah ini. :

Tabel 2.2 Anak Huruf

Nama	Bunyi	Aksara	Posisi
Datas	An	=	Diatas
Ulan	I	ı	Diatas
Ulan	e	´	Diatas
Bicek	e	´	Diatas
Tekelubang	ng	-	Diatas
Rejunjung	r	ˆ	Diatas
Tekelungau	au	˘	Diatas
Bitan	u	-	Dibawah
Bitan	o	,	Dibawah
Tekelingai	ai	ı	Disamping
Keleniah	ah	-	Disamping
Nengen	mati	ı	Disamping

2.4 Aplikasi *Mobile*

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya

komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain – lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu:

1. Ukuran yang kecil : Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary (RAM)* dan *secondary (disk)*.
3. Daya proses yang terbatas : Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
4. Mengonsumsi daya yang rendah : Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*.
5. Kuat dan dapat diandalkan : Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas : Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek : Perangkat – perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.5 *Android*

(H,Nazruddin Safaat 2015) *Android* adalah istilah dalam bahasa inggris yang berarti “Robot yang menyerupai manusia”. Logo *android* sendiri dicerminkan seperti sebuah robot berwarna hijau, yang mengacu kepada arti kata *Android*. *Android* adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunaanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi – aplikasi yang tersedia pada

device. Di dunia personal komputer, sistem operasi yang banyak dipakai adalah *Windows*, *Mac*, dan *Linux*.

Android dikembangkan bersama oleh perusahaan-perusahaan yang tergabung dalam sebuah konsorsium bernama *Open Handset Alliance (OHA)*. *OHA* dipimpin oleh *Google* dan didirikan bersama dengan 34 perusahaan lainnya, dengan tujuan untuk mengembangkan teknologi *mobile device*, semikonduktor, pembuatan aplikasi, komersialisasi, dan *mobile operator*.

2.6 Versi dan Jenis-Jenis *Android*

Pengembangan Android dimulai dengan berdirinya Android, Inc. Pada Oktober 2003 dengan tujuan membuat *mobile device* yang lebih *smart* untuk menyaingi *Symbian* dan *Windows Mobile* yang populer saat itu (*iPone* dan *Blackberry* belum diliris). Pada tahun 2005, Android Inc. Diakuisisi oleh *google*. Versi dan jenis-jenis *android* dapat dilihat pada table 2.3 berikut ini :

Tabel 2.3 Versi dan Jenis – Jenis *Android*

Versi	Nama	Rilis	Catatan
1.0	-	23 September 2008	Android pertama, hanya untuk <i>smartphone</i>
1.1	-	9 Februari 2009	
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama
1.6	<i>Donut</i>	15 September 2009	
2.0-2.1	<i>Eclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	

Tabel 2.3 Versi dan Jenis – Jenis *Android* (Lanjutan)

2.2	<i>Froya</i> (<i>Frozen</i> <i>Yogurt</i>)	20 Mei 2010	
2.3	<i>Gingerbread</i>	6 Desember 2010	Masih banyak digunakan di <i>smartphone</i> jenis lama
3.0	<i>Honeycomb</i>	22 Februari 2011 (3.0)	Hanya untuk tablet
3.2		10 Mei 2011 (3.1)	
		15 Juli 2011 (3.2)	
4.0	<i>ICS (Ice</i> <i>Cream</i> <i>Sandwich)</i>	19 Oktober 2011	Smartphone dan Tablet
4.1-4.3	<i>Jelly Bean</i>	9 Juli 2012 (4.1)	<i>Update</i> untuk memperbaiki dan menambah fitur-fitur <i>ICS</i>
		13 November 2012 (4.2)	
		24 Juli 2013 (4.3)	
4.4	<i>Kit Kat</i>	-	Diumumkan 3 September 2013
5.0	<i>Lollipop</i>		

2.7 *PHP*

(H,Priyanto dan Jauhari K.K 2017) *Hypertext Preprocessor (PHP)* adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*.

PHP juga dapat diintegrasikan dengan HTML, *Javascript*, *JQuery*, *Ajax*. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan

file bertipe HTML. Dengan Menggunakan PHP bisa membuat *website powerful* yang dinamis dengan disertai manajemen database-nya. Selain itu penggunaan PHP dapat jalan di banyak *platform*.

2.8 *HTML (Hyper Text Markup Language)*

(H,Priyanto dan Jauhari K.K 2017) *Hypertext Markup Language* (HTML) adalah bahasa *standard* yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan dengan HTML yaitu :

- Mengatur tampilan dari halaman *web* dan isinya.
- Membuat tabel dalam halaman *web*.
- Memublikasikan halaman *web* secara *online*.
- Membuat *Form* yang bisa di gunakan untuk menangani registrasi dan transaksi via *web*.
- Menambahkan objek-objek seperti citra, audio, video, animasi, java applet dalam halaman *web*.
- Menampilkan area gambar (canvas) di browser.

2.9 *MySQL*

(H,Priyanto dan Jauhari K.K 2017) *MySQL* adalah salah satu aplikasi DBMS(*Database Management System*) yang sudah sangat banyak digunakan oleh para pemrogram aplikasi *web*. Kelebihan dari MySQL adalah gratis, handal, selalu di *update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah.

2.10 XAMPP

(H,Priyanto dan Jauhari K.K 2017), *XAMPP* ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari program *MySQL database*, *Apache HTTP Server*, dan penerjemah ditulis dalam bahasa pemrograman *PHP* dan *Perl*.

Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia di bawah *GNU (General Public License)* dan bebas, adalah mudah untuk menggunakan *web server* yang dapat melayani tampilan halaman *web* yang dinamis.

2.11 APK (Android Package)

APK adalah paket aplikasi Android (*Android PacKage*). APK umumnya digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat Android. APK pada dasarnya seperti zip file, karena berisi dari kumpulan file. APK dapat diperoleh melalui berbagai metode, seperti menginstal sebuah aplikasi melalui #Market, *download* dari sebuah situs web, atau membuat sendiri dengan bahasa *Java*. Jika Anda memiliki file APK pada komputer Anda, dan ingin menginstalnya pada telepon Anda, Anda dapat menjalankan perintah ‘adb install apkname.apk’ untuk install aplikasi melalui USB ke telepon Anda.

File APK berada di perangkat Android di direktori */data/app/filename.apk*. Direktori ini biasanya tidak dapat diakses, untuk melindungi aplikasi berbayar, kecuali #rooted #ROM digunakan di telepon. Pada *rooted ROM*, sangat mungkin untuk menyalin aplikasi ke folder ini secara manual dengan menggunakan perintah ‘adb push’ atau semacamnya. Setelah

sebuah file berada di direktori `/data/app`, maka aplikasi sudah terpasang pada telepon.

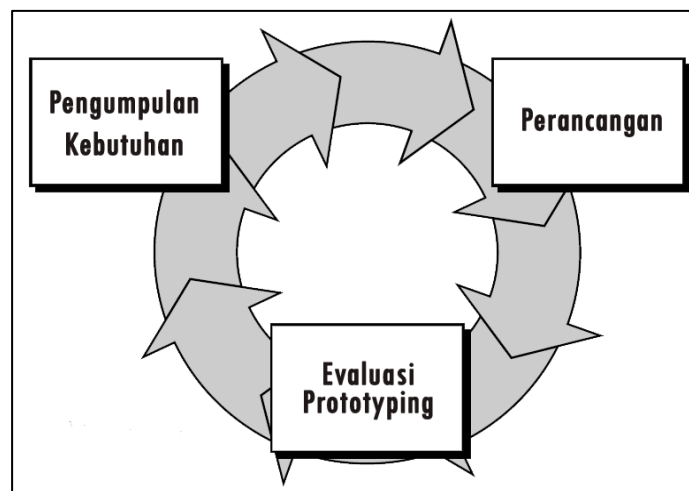
Selain itu dimungkinkan juga untuk menginstal aplikasi ke partisi `#sistem` dari telepon yang sudah di-root. Cara ini memiliki keuntungan yaitu aplikasi tersebut tidak dapat dihapus dari perangkat, sehingga berguna untuk keamanan/anti-pencurian aplikasi seperti *Wave Secure* dan *Mobile Defense*. `#Sistem` aplikasi ini akan disimpan dalam `/system/app`. Aplikasi dalam folder ini biasanya bagian dari sistem operasi Android, dan sebaiknya jangan diubah atau dihapus kecuali Anda mengetahui apa yang Anda lakukan.

File APK dapat dihapus dengan banyak cara, antara lain: menggunakan aplikasi `#Market` (jika aplikasi diinstal dari Market), perintah `'uninstall adb apkgname.apk'`, melalui menu *Settings | Applications | Manage Applications*, atau dengan menghapus secara manual file-file dari direktori `/data/app`. Namun biasanya metode yang sering digunakan adalah melalui market atau menu aplikasi.

2.12 Metode Pengembangan Sistem

2.12.1 Metode *Prototype*

(S,Rosa A. dan M.Shalahuddin 2016) 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 2.2 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.

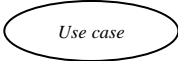
2.13 UML (*Unified Modelling Language*)

(S, Rosa A. dan M. Shalahuddin 2016) UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

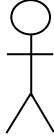

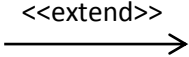
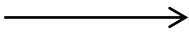
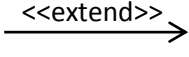
2.13.1 Use Case Diagram

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih actor dan use case juga digunakan mengetahui fungsi apa saja yang terdapat didalam sistem. Adapun simbol dari *use case* diagram terdapat pada tabel 2.4 berikut ini:

Tabel 2.4 Simbol *Use Case* Diagram

Nama Komponen	Keterangan	Simbol
<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case	




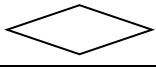

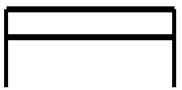
Tabel 2.4 Simbol *Use Case Diagram* (lanjutan)

<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi. walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda di awal frase nama aktor	
<i>Association</i>	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor	
<i>Extend</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan.	
<i>generalization</i>	Hubungan generalisasi dan spesialisasi antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.	
<i>include</i>	Use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan	

2.13.2 Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Adapun symbol – symbol dari activity diagram terdapat pada tabel 2.5 berikut ini :



Tabel 2.5 Simbol *Activity Diagram*

Simbol	Keterangan
	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan dengan kata kerja
	Asosiasi percabangan dimana jika ada pilihan
	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


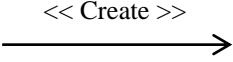
2.13.3 Sequence Diagram

Sequence diagram secara grafis menggambarkan bagaimana object berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah *use case* atau proses. Adapun simbol-simbol dari *sequence* diagram terdapat pada tabel 2.6 berikut:

Tabel 2.6 Simbol *Sequence Diagram*

Nama Komponen	Komponen	Simbol
<i>Lifeline</i>	Menyatakan kehidupan suatu objek	
<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya	

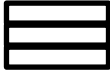


Tabel 2.6 Simbol *Sequence Diagram* (Lanjutan)

<i>Object</i>	Menyatakan objek yang berinteraksi	
Pesan tipe create	Menyatakan suatu objek membuat objek yang lain	

2.13.4 Class Diagram

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. Adapun simbol-simbol dari *Class* diagram terdapat pada tabel 2.7 berikut:

Tabel 2.6 Simbol *Class Diagram*

Nama Komponen	Komponen	Simbol
<i>Class</i>	Kelas pada struktur sistem	
<i>Association</i>	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	

2.14 Skripsi Rancang Bangun Media Pembelajaran Pembuatan Film Indie Untuk Filmmaker Pemula Berbasis Mobile

(Permadi, Bambang, 2015) Skripsi ini merupakan referensi dalam pembuatan skripsi ini. Didalam skripsi ini diterangkan bagaimana cara

membangun media pembelajaran Pembuatan Film Indie Untuk Filmmaker Pemula

2.15 ATOM

Atom adalah sebuah text editor yang memiliki lisensi open source yang tersedia untuk platform OS X, Linux dan Windows. Atom ini dibuat oleh GitHub dan di klaim sebagai text editor yang bisa di custom dengan merubah file configurasinya. Atom ini mirip dengan salah satu text editor favorit saya yaitu Sublime Text, karena Atom ini memang dibuat dengan menggunakan Sublime Text sebagai referensinya

2.16 JQuery

(H,Priyanto dan Jauhari K.K 2017) JQuery adalah kumpulan fungsi-fungsi Javascript yang sudah dibentuk suatu objek, *JQuery* adalah suatu Javascript *Library* karena JQuery berada di atas JavaScript. Sehingga penggunaan JQuery ini bisa dikategorikan sebagai suatu *library* yang nantinya hanya menggunakan fungsi-fungsi di dalam library tersebut.

2.17 Kamus Data

(S,Rosa A. dan M.Shalahuddin 2016) Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak dapat dipahami secara umum.Kamus data berisi nama dari data dan informasi tambahan seperti tipe data,nilai data.

2.18 Skripsi Rancang Bangun Media Pembelajaran Basis Data Terdistribusi Berbasis Multimedia

(S, Ricard D. 2015) Skripsi ini merupakan referensi dalam pembuatan skripsi ini. Didalam skripsi ini diterangkan bagaimana cara membangun media pembelajaran Basis Data Terdistribusi.