

# Rancang Bangun Aplikasi Buku “Dongeng” - Kumpulan Cerita Rakyat Interaktif Berbasis iOS

Romin Adi Santoso, Dwi Sunaryono, dan Isye Arieshanti

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

*e-mail: dwi@its-sby.edu*

**Abstrak**—Kesukaan anak muda pada saat ini terhadap cerita rakyat lokal semakin pudar. Cerita rakyat dari luar negeri lebih disukai misalnya saja putri salju atau Cinderella. Lemahnya pamor cerita rakyat lokal ini mungkin dikarenakan publikasi tentang cerita rakyat lokal yang kalah dengan cerita yang berasal dari luar negeri. Selain dikarenakan cerita luar negeri yang lebih digandrungi anak-anak muda saat ini, dikarenakan permainan digital yang perkembangannya yang sangat pesat. Salah satu media permainan yang dianggap menarik banyak anak di dunia ini adalah iPad. Melihat dari permasalahan akan kurangnya minat anak-anak terhadap cerita rakyat lokal dan meningkatnya popularitas iPad sebagai *toys of the year* versi majalah PCWorld, maka munculah gagasan artikel untuk membuat aplikasi berbasis iPad berjudul “Dongeng” yang merupakan kumpulan cerita rakyat Indonesia interaktif. Interaktif di sini dikarenakan pengguna bisa berinteraksi dengan karakter maupun lingkungan dalam cerita rakyat yang dipilih. Selain itu, interaksi dari pengguna bisa menentukan alur cerita dari cerita itu sendiri. Pengguna akan diajak berpikir dalam menentukan alur cerita agar tidak salah dalam memilih alur ceritanya. Pemilihan alur cerita bisa berdasarkan event berupa mini-game maupun pertanyaan. Aplikasi ini nantinya diharapkan dapat lebih mengenalkan cerita rakyat lokal kepada anak-anak baik dari dalam maupun luar negeri.

**Kata kunci**—Cerita Rakyat, iPad, Permainan.

## I. PENDAHULUAN

PADA jaman sekarang ini, kecintaan anak-anak terhadap cerita rakyat lokal semakin berkurang. Mereka lebih menyukai cerita dari luar Indonesia seperti Amerika dan Jepang. Terdapat beberapa penyebab yang dapat disimpulkan mengapa cerita rakyat lokal kurang diminati oleh anak-anak saat ini. Salah satunya adalah karena sudah sangat jarang orang tua yang mau meluangkan waktunya untuk menceritakan cerita rakyat kepada anak-anak mereka. Selain itu, popularitas cerita luar yang didukung kuat oleh media *digital* sedangkan cerita rakyat Indonesia yang hanya sebatas kertas ataupun media *digital* yang bisa dibilang sudah tidak mengikuti jaman dan kurang dikemas menarik.

Saat ini, perkembangan teknologi *tablet* yang sudah berkembang pesat membuat banyak anak muda saat ini yang lebih banyak menghabiskan waktu dengan perangkat *tablet* mereka seperti iPad atau Android. Bahkan karena tren perangkat *tablet* yang sangat populer membuat kaum muda saat ini cerita-cerita yang berasal dari luar sekalipun mulai

ditinggalkan orang. Jika ketenaran cerita yang berasal dari luar sudah mulai menurun, otomatis pamor produk lokal juga pasti menurun juga. Hal ini bisa dibilang cukup memprihatinkan karena produk lokal kita yang semakin kalah bersaing di pasar Indonesia ini sendiri.

Berdasarkan hal tersebut, dalam artikel ini dikembangkan aplikasi cerita rakyat lokal bernama **Dongeng**. Aplikasi ini akan berjalan di lingkungan sistem operasi iOS. Meskipun aplikasi ini berjalan di sistem operasi iOS, aplikasi ini hanya mendukung resolusi iPad. Aplikasi ini akan memanfaatkan kakas kerja UIKit, Cocos2d V2, Cocoa Touch, Box2d, Game Center, dan Store Kit.

## II. TINJAUAN PUSTAKA

### A. Cocos2d

*Engine* utama yang digunakan untuk membangun aplikasi **Dongeng** ini adalah Cocos2d [1]. Cocos2d merupakan sebuah *engine Open Source* yang dibangun oleh komunitas. Bahasa yang digunakan oleh Cocos2d adalah bahasa asli yang didukung Apple. Karena menggunakan bahasa asli, maka aplikasi yang dibangun dengan Cocos2d memiliki kecepatan yang lebih cepat dari *engine* lain seperti Adobe Flash.

### B. Box2d

Box2d [1] adalah sebuah *engine* Fisika yang dibangun dengan bahasa C++. Terdapat berbagai macam fitur yang diberikan oleh Box2d seperti deteksi tabrakan kontinyu, berbagai macam bentuk fisika untuk satu objek/gambar, pengaturan besar pantulan dan kasar atau licinnya objek. Selain itu terdapat berbagai persendian seperti: *revolute, prismatic, distance, pulley, gear, mouse joint*, dan berbagai macam sendi lainnya. Pada Box2d juga terdapat motor untuk menggerakkan sendi tersebut. Serta batasan sudut yang bisa dicapai.

Box2d yang diintegrasikan dengan Cocos2d ini nantinya akan digunakan untuk menambah efek realita dalam aplikasi **Dongeng** ini.

### C. Kerangka Kerja UIKit

Kakas kerja UIKit merupakan kakas kerja dasar dalam setiap pembuatan aplikasi berbasis iOS. UIKit menyediakan objek-objek kunci yang diperlukan untuk menjalankan

aplikasi, untuk menangkap masukan dari pengguna (*touching*), dan untuk menampilkan konten pada layar.

Banyak interaksi yang terjadi antara sistem dan sebuah aplikasi yang berjalan. Banyak dari interaksi-interaksi tersebut yang diatur secara otomatis oleh infrastruktur UIKit. Contoh *event* adalah misalnya ketika pengguna keluar dari aplikasi, pengembang bisa memanfaatkan UIKit ini untuk melakukan beberapa eksekusi seperti menyimpan beberapa data ketika aplikasi tertutup atau yang lainnya. Komponen-komponen seperti UIAlertView, UITextField, dan lainnya juga merupakan fitur yang ada di dalam UIKit. Dengan kata lain, UIKit sebenarnya mirip dengan kaskas kerja .NET yang dapat dijumpai di Microsoft.

**D. Kerangka Kerja GameKit**

Orang-orang sangat menyukai permainan. Tidak terkecuali permainan yang ada di AppStore. Permainan merupakan turunan dari aktifitas sosial. Pada beberapa waktu, interaksi sosial adalah bagian dari permainan itu sendiri seperti pada saat permainan menyediakan fitur *multiplayer*. Tetapi, bahkan untuk permainan yang dibuat untuk *single player*, pemain ingin untuk melihat prestasi-prestasi yang telah dicapai oleh pemain lain.

Karena *social gaming* merupakan hal penting dalam bagian pengalaman bermain permainan, Apple mendukung *social gaming* secara langsung dengan layanan Game Center [2]. Game Center memperbolehkan perangkat iOS pengguna untuk tersambung ke layanan Game Center dan bertukar informasi.

Terdapat berbagai macam fitur yang ditawarkan oleh GameKit [2], tetapi pada aplikasi **Dongeng** ini fitur yang digunakan adalah pencapaian *achievement* ketika pengguna telah menyelesaikan beberapa. Setiap *achievement* akan menambahkan *score* kepada akun Game Center yang dimiliki pengguna.

**E. IAP**

IAP merupakan fitur pada iOS yang berguna untuk transaksi konten-konten digital di dalam aplikasi berbasis iOS [3]. Dengan IAP, dapat dilakukan transaksi-transaksi sebagai berikut, contohnya:

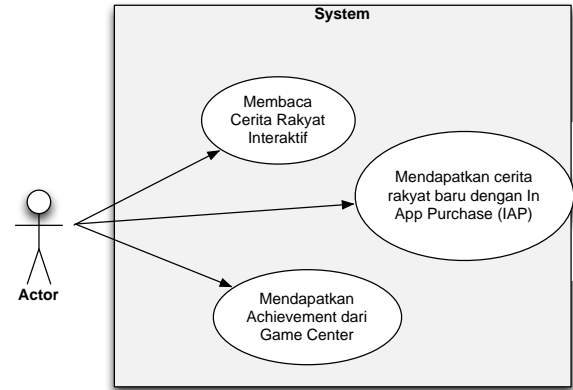
1. Menjual buku digital dan foto-foto.
2. Menambah *game levels*.
3. Berlangganan majalah digital atau surat kabar digital.

**III. ANALISIS DAN PERANCANGAN SISTEM**

Analisis kebutuhan perangkat lunak dilakukan didasarkan pada perumusan permasalahan yang ada pada artikel ini. Dari hasil analisis tersebut kemudian dilanjutkan pada perancangan perangkat lunak yang bisa memenuhi semua kebutuhan fungsional. Perancangan perangkat lunak ini menggunakan pendekatan perancangan berorientasi objek yang direpresentasikan dengan UML (*Unified Modelling Language*).

**A. Analisis Perangkat Lunak**

Analisis yang dilakukan pada artikel ini meliputi deskripsi secara umum terhadap perangkat lunak yang dibangun, spesifikasi dari kebutuhan apa saja yang diperlukan



Gambar 1. Diagram kasus penggunaan

perangkat lunak tersebut, analisis aktor yang menjadi pengguna perangkat lunak hingga analisis kasus penggunaan.

Dalam artikel ini, dibangun sebuah aplikasi kumpulan cerita rakyat interaktif berbasis iOS yang bernama **Dongeng**. Aplikasi ini merupakan buku cerita rakyat digital yang berisikan 5 buah cerita rakyat lokal. Tujuan dari pembuatan tugas akhir ini adalah untuk merancang dan membangun suatu aplikasi termasuk kategori buku *digital* interaktif berbasis iOS yang bisa menjadi sarana hiburan dan promosi cerita rakyat Indonesia yang praktis.

Daftar kebutuhan fungsional dari sistem yang dibangun dijelaskan dengan diagram kasus penggunaan pada Gambar 1 dan rincian kasus penggunaan.

- *Kasus Penggunaan “Membaca Cerita Rakyat Interaktif”*  
Kasus penggunaan “Membaca Cerita Rakyat Interaktif” merupakan kasus penggunaan yang diakses secara langsung oleh pengguna (anak-anak).
- *Kasus Penggunaan “Mendapatkan Cerita Rakyat Baru dengan IAP”*  
Kasus penggunaan “Mendapatkan Cerita Rakyat Baru dengan IAP” merupakan kasus penggunaan yang diakses secara langsung oleh pengguna (orang tua).
- *Kasus penggunaan “Mendapatkan Achievements dari Game Center”*  
Kasus penggunaan “Mendapatkan Achievements dari Game Center” merupakan kasus penggunaan yang diakses secara langsung oleh pengguna (anak-anak).

**B. Arsitektur Sistem**

Berikut ini merupakan arsitektur yang digunakan untuk membangun aplikasi **Dongeng**. Secara keseluruhan sistem yang menaungi **Dongeng** dibagi menjadi enam buah lapisan utama. Lapisan tersebut adalah Lapisan Antarmuka (*Presentation Layer*), Lapisan Model Data (*Data Model Layer*), Lapisan Logika Bisnis (*Bussines Logic Layer*), dan *Core Services Layer*. Pada Gambar 2 ditunjukkan bagaimana keterhubungan antar lapisan.

**C. Perancangan Diagram Kelas**

Lapisan ini berfungsi sebagai pengatur proses berjalannya aplikasi. Lapisan ini juga berisi fungsi-fungsi yang akan dipanggil oleh lapisan antarmuka. Logika bisnis yang dipakai di sini adalah satu *controller* untuk satu *view*

dikarenakan setiap halaman cerita memiliki logika pengerjaan yang berbeda.

Meskipun logika yang dipakai di setiap halaman / *scene* berbeda-beda, setiap halaman tersebut memiliki kemiripan struktur kode. Struktur kode tersebut akan dibahas di bab IV (implementasi). Selain struktur kode kelas setiap halaman, untuk mempermudah dan mempersingkat pembuatan setiap halaman yang ada, dibuat sebuah kelas *singleton* bernama *game manager*. Kelas ini memiliki banyak fungsi yang nantinya membantu kelas *controller* pada setiap halaman. Pada setiap kelas *controller* halaman akan memiliki beberapa fungsi yang selalu dipakai. Kelas *game manager* ini akan menampung semua fungsi tersebut agar tidak terjadi penulisan kode yang *redundant*.

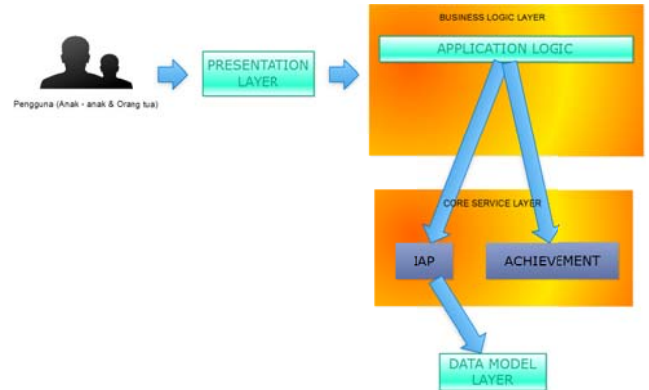
Gambar 3, adalah rancangan kelas controller dari setiap halaman cerita. Gambar 4 adalah rancangan kelas *GameManager*. Kelas ini berfungsi untuk membantu pelembaran data antar halaman. Selain itu, kelas ini juga bertujuan untuk tempat penyimpanan *resource* dan fungsi-fungsi yang dapat digunakan berulang-ulang.

#### IV. IMPLEMENTASI

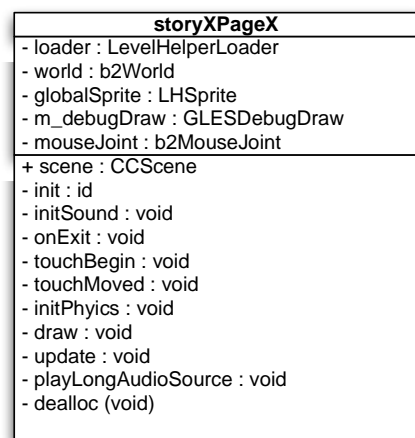
Dalam aplikasi **Dongeng** ini logika bisnis berfungsi sebagai *controller* dari *view*. Seperti yang telah dijelaskan pada bab sebelumnya, setiap halaman cerita memiliki satu buah *controller*. Terdapat pula sebuah kelas *singleton* yang berfungsi untuk membantu kelas *controller* tersebut. Kelas ini akan menampung fungsi-fungsi yang sama pada setiap kelas *controller*. Selain itu, kelas *singleton* ini berfungsi juga untuk menyimpan keadaan apakah suara narasi diaktifkan atau tidak.

##### 1) Membuat Scene

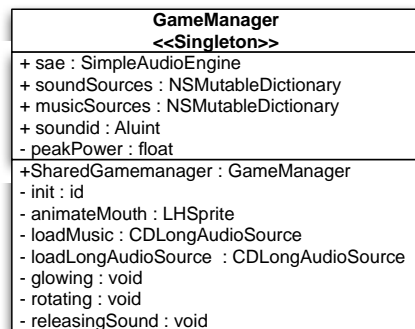
Membuat *Scene* bertujuan sebagai wadah untuk menampung *layer* serta semua *nodes* di dalamnya. Kode 1 berisi bagaimana sebuah *scene* (**CCScene**) dan sebuah *layer* (**CCLayer**) diinstansiasi. Baris bertama merupakan proses instansiasi dari **CCScene** dengan memanggil [**CCScene node**]. Pemanggilan [**CCScene node**] tersebut merupakan kependekan dari [[[**CCScene alloc**] **init**] **autorelease**]. *Layer* tersebut kemudian ditambahkan ke *scene* dengan cara memanggil fungsi *addChild* pada *scene*.



Gambar 2. Arsitektur sistem



Gambar 3. Rancangan kelas diagram halaman cerita



Gambar 4. Rancangan kelas gamemanager

```

+(id) scene {

    // 'scene' is an autorelease object.
    CCScene *scene = [CCScene node];
    // 'layer' is an autorelease object.
    story5page7 *layer = [story5page7 node];
    // add layer as a child to scene
    [scene addChild: layer];
    return scene; }

```

Kode 1. Membuat *scene* baru

```

-(id) init
{
    if (self=[super init]) {
        self.isTouchEnabled = YES;
        self.isAccelerometerEnabled = YES;
        [self initPhysics];
        ceknext = 0;
        loader = [[LevelHelperLoader alloc]
initWithContentOfFile:@"story 5 level 7"];
        [loader addObjectToWorld:world
cocos2dLayer:self];
        [loader createPhysicsBoundaries:world];
        LHSprite* next = [loader
spriteWithUniqueName:@"next-HD"];
        [next registerTouchBeganObserver:self
selector:@selector(touchBegin:)];
        LHSprite* prev = [loader
spriteWithUniqueName:@"previous-HD"];
        [prev registerTouchBeganObserver:self
selector:@selector(touchBegin:)];
        NSArray* arr = [loader spritesWithTag:PAGE3EMBER];
        for (LHSprite* l in arr) {
            [self skewing:l];
            [l registerTouchBeganObserver:self
selector:@selector(touchBegin:)];
        }
        NSArray* arr1 = [loader
spritesWithTag:PAGE3BJUPUTIH];
        for (LHSprite* l in arr1) {
            [self skewing:l];
            [l registerTouchBeganObserver:self
selector:@selector(touchBegin:)];
        }

        panah = [loader
spriteWithUniqueName:@"panah"];
        panah1 = [loader
spriteWithUniqueName:@"panah_1"];
        [[GameManager
sharedGameManager]glowing:panah];
        [[GameManager
sharedGameManager]glowing:panah1];

        cekcerita = 0;
        [self initsound];
        [self scheduleUpdate];
    }
    return self;
}

```

Kode 2. Fungsi *init*

## 2) Inisialisasi

Ketika *layer story5page7* dipanggil, fungsi *init* dipanggil. Pada fungsi inilah semua *nodes* ditambahkan ke dalam *layer* seperti pada Kode 2.

## 3) Inisiasi Physics

Inisiasi *physic* bertujuan untuk menyiapkan lingkungan *physic* untuk objek-objek Box2d seperti besaran gravitasi dan mendefinisikan batas-batas layar. Fungsi inisiasi *physic* sendiri bernama *initPhysics* seperti pada Kode 3.

```

-(void) initPhysics
{
    CGSize s = [[CCDirector sharedDirector]
winSize];
    b2Vec2 gravity;
    gravity.Set(0.0f, -10.0f);
    world = new b2World(gravity);
    // Do we want to let bodies sleep?
    world->SetAllowSleeping(true);
    world->SetContinuousPhysics(true);
    m_debugDraw = new GLESDebugDraw( PTM_RATIO
);

    world->SetDebugDraw(m_debugDraw);
    uint32 flags = 0;
    flags += b2Draw::e_shapeBit;
    m_debugDraw->SetFlags(flags);
    // Define the ground body.
    b2BodyDef groundBodyDef;
    groundBodyDef.position.Set(0, 0);
    b2Body* groundBody = world-
>CreateBody(&groundBodyDef);
    // Define the ground box shape.
    b2EdgeShape groundBox;
    // bottom
    groundBox.Set(b2Vec2(0,0),
b2Vec2(s.width/PTM_RATIO,0));
    groundBody->CreateFixture(&groundBox,0);
    // top
    groundBox.Set(b2Vec2(0,s.height/PTM_RATIO),
b2Vec2(s.width/PTM_RATIO,s.height/PTM_RATIO));
    groundBody->CreateFixture(&groundBox,0);
    // left
    groundBox.Set(b2Vec2(0,s.height/PTM_RATIO),
b2Vec2(0,0));
    groundBody->CreateFixture(&groundBox,0);
    // right
    groundBox.Set(b2Vec2(s.width/PTM_RATIO,s.hei
ght/PTM_RATIO), b2Vec2(s.width/PTM_RATIO,0));
    groundBody->CreateFixture(&groundBox,0);
}

```

Kode 3. Fungsi *init physic*

```

-(void) update: (ccTime) dt
{
    int32 velocityIterations = 8;
    int32 positionIterations = 1;
    world->Step(dt, velocityIterations,
positionIterations);

    for (b2Body* b = world->GetBodyList(); b; b =
b->GetNext())
    {
        if (b->GetUserData() != NULL)
        {
            CCSprite *myActor = (CCSprite*)b-
>GetUserData();
            if(myActor != 0)
            {
                myActor.position =
[LevelHelperLoader metersToPoints:b-
>GetPosition()];
                myActor.rotation = -1 *
CC_RADIANS_TO_DEGREES(b->GetAngle());
            }}
}

```

Kode 4. Fungsi *update*

## 4) Perulangan Permainan

Perulangan ini bertujuan untuk membuat gambar bergerak karena pada perulangan ini akan melakukan *update frame* pada aplikasi. Fungsi perulangan yang bernama *update* ini akan dipanggil setiap pergantian *frame*. Kode 4 adalah isi dari fungsi *update*.

V. PENGUJIAN DAN EVALUASI

Pengembangan perangkat lunak dilakukan dengan dua pengujian yaitu pengujian fungsionalitas untuk mengecek kebenaran dan pengujian secara subjektif ketika aplikasi digunakan oleh pengguna secara nyata. Kedua pengujian tersebut menggunakan pengujian dengan *blackbox*.

A. Pengujian Subjektif

Uji coba pengguna dilakukan untuk mengetahui apakah aplikasi **Dongeng** ini sudah tepat sasaran atau belum. Selain itu, uji coba pengguna ini dapat sebagai sarana untuk mengetahui apakah aplikasi **Dongeng** ini sudah menarik bagi yang membacanya. Pengujian ini dilakukan pengguna dengan metode *blackbox testing*.

Pengujian dilakukan kepada beberapa pengguna anak-anak. Pengujian aplikasi ini dilakukan dengan cara para penguji mencoba 2-3 cerita yang ditawarkan di dalam aplikasi **Dongeng** ini.

Setelah penguji melakukan pengujian dengan membaca cerita secara acak, penguji mengisi sebuah lembar kuisioner yang dibimbing oleh orang tua penguji. Pertanyaan pada kuisioner berbentuk pilihan ganda dan wajib diisi semua oleh penguji.

Nilai akhir merupakan nilai rata-rata dari tingkat kemudahan pengguna dalam menggunakan aplikasi, tingkat kepuasan pada antarmuka aplikasi, dan tingkat ketertarikan terhadap konten. Setelah mendapatkan nilai akhir tersebut (3,633 dari skala 5) maka dapat ditarik kesimpulan bahwa aplikasi **Dongeng** ini sudah tepat sasaran karena sudah mencapai nilai lebih dari 60%.

B. Analisis Hasil Pengujian

Setelah mendapatkan daftar penguji yang didapatkan acak tersebut, maka didapatkan hasil observasi. Hasil observasi tersebut berupa kuisioner yang diisi oleh orang tua masing-masing anak. Tabel 1 menunjukkan hasil rekapitulasi pengujian pembuatan rancangan basis data.

Nilai akhir dari masing-masing aspek terlihat pada Tabel 1. Berdasarkan nilai tersebut, aplikasi ini memiliki rentang nilai antara 3 sampai 4,1 dengan skala penilaian 0 hingga 5. Dari nilai tersebut dapat disimpulkan aplikasi **Dongeng** ini cukup memuaskan pengguna dengan rentang usia tersebut.

C. Uji Coba Performa Sistem

Uji coba performa sistem ini bertujuan untuk mengetahui apakah seberapa besar aplikasi **Dongeng** dalam menggunakan sumber daya seperti memori dan berapa persen proses pada CPU.

Uji coba alokasi penggunaan memori adalah uji coba yang ditujukan untuk mengetahui alokasi memori yang digunakan oleh aplikasi ketika berjalan. Uji coba ini dilakukan dengan membuka aplikasi dan memulai membuka salah satu cerita.

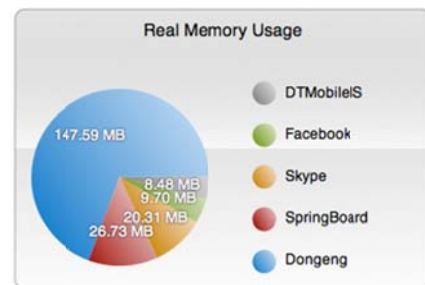
Jumlah pemakaian memori ketika aplikasi berjalan dapat dilihat pada Gambar 5. Dari Gambar 5 tersebut dapat diketahui bahwa aplikasi menggunakan memori sebesar 147,59 MB. Hasil uji coba alokasi penggunaan memori ini dilakukan pada perangkat iPad 2.

Uji coba alokasi penggunaan CPU adalah uji coba yang ditujukan untuk mengetahui pemakaian CPU ketika aplikasi dijalankan. Uji coba ini dilakukan dengan membuka aplikasi

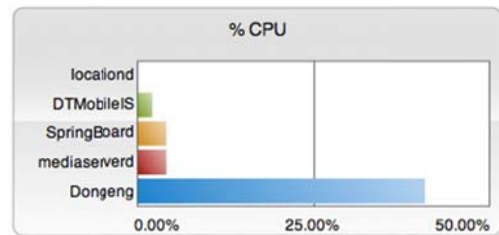
dan membuka cerita. Hasil dari uji coba ini bisa dilihat pada Gambar 6. Pada Gambar 6 tersebut, ditunjukkan bahwa alokasi penggunaan CPU hampir mencapai 50%. Uji coba alokasi penggunaan CPU ini menggunakan perangkat iPad 2.

Tabel 1. Hasil penilaian uji coba pengguna

No	Parameter	Penilaian					Nilai
		1	2	3	4	5	
1	Tingkat kemudahan penggunaan		2		5	3	3,9
2	Tingkat kepuasan pada antarmuka aplikasi		2		3	5	4,1
4	Tingkat ketertarikan pada konten cerita		3	5	2		2,9
Nilai akhir							3,6333



Gambar 5. Alokasi memori Dongeng



Gambar 6. Alokasi penggunaan cpu pada Dongeng

VI. KESIMPULAN

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba yang dilakukan, kesimpulan yang dapat diambil berdasar rumusan masalah yang ada adalah sebagai berikut ini.

1. Aplikasi **Dongeng** ini menarik minat anak-anak pada umur 6-8 tahun. Sedangkan untuk anak yang berusia lebih tua cenderung kurang memiliki ketertarikan dengan konten yang disediakan. Hal ini ditunjukkan pada hasil uji coba pengguna.
2. Untuk anak berumur kurang dari 6 masih dibutuhkan bimbingan dari orang tua. Hal ini dikarenakan anak-anak tersebut belum dapat membaca dengan baik. Hal ini dijelaskan pada Tabel 1.

Dalam pembuatan Artikel ini, terdapat beberapa saran dan perbaikan bagi sistem yang telah diimplementasikan sebagai berikut ini.

1. Aplikasi **Dongeng** ini dapat diperbaiki dengan menambahkan variasi animasi, permainan, dan percabangan cerita sehingga bisa menjadi aplikasi yang lebih menarik bagi anak-anak yang berusia lebih dari 8 tahun.

2. Untuk meningkatkan efisiensi alokasi memori, untuk kedepannya bisa dicari cara untuk membuat memori yang digunakan lebih sedikit.

#### UCAPAN TERIMA KASIH

Penulis R.A.S. mengucapkan terima kasih kepada Tuhan Yang Maha Esa, kedua orangtua dan keluarga penulis, dosen pembimbing, seluruh dosen Teknik Informatika ITS, kerabat dekat, PT. Angkasa Pura, serta berbagai pihak yang telah membantu penulis dalam menyelesaikan penelitian ini.

#### DAFTAR PUSTAKA

- [1] Nathan Burba, Cocos2d for iPhone Game Development Cookbook, 1st ed., Nathan Burba, Ed. Birmingham, India: Packt Publishing, 2011.
- [2] Apple. (2012, September) Apple. [Online]. [http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/GameKit\\_Guide/Introduction/Introduction.html](http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/GameKit_Guide/Introduction/Introduction.html)
- [3] Apple. (2013, January) In-App Purchase for Developer. [Online]. <https://developer.apple.com/in-app-purchase/In-App-Purchase-Guidelines.pdf>