KALBIS Scientia
Jurnal Sains dan Teknologi

ISSN 2356 - 4393

Aplikasi Permainan Untuk Pembelajaran Algoritma Bubble Sort

Charlie[1)], Alfa Ryano Yohanis[2)]

Teknik Informatika, Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav. 22, Jakarta Timur 13210
[1)]*Email: charli3chai@yahoo.com*
[2)]*Email: alfa.yohannis@kalbis.ac.id*

Abstract: *In computer science, sorting algorithms is one of the topics to learn how to sort the data programmatically. To help a person in learning sorting algorithms, students can take advantage of the learning applications in the form of a game. The application can provide motivation and effectiveness in learning sorting algorithms. This study aims to produce a game application which explains the steps of bubble sort algorithm involves user interaction in the form of a game. Applications is built using the Java programming language and run on the Android operating system that only require a smart phone device to run the application. This research method is done by analyzing the related theories, applications and similar studies. Requirements derived from the analysis are used for application development. Application development using the waterfall model which comprises the step of the analysis phase, design, coding and testing. Results of the research is a computer game application that can be used to study bubble sort algorithm which is expected to help someone in understanding the steps of bubble sort algorithm.*

Keywords: *Android, bubble sor, computer game, Java*

Abstrak: *Dalam ilmu komputer, algoritma pengurutan merupakan salah satu topik yang mempelajari bagaimana mengurutkan data secara program. Untuk membantu seseorang dalam mempelajari algoritma pengurutan, pelajar dapat memanfaatkan aplikasi pembelajaran dalam bentuk permainan. Aplikasi tersebut dapat memberikan motivasi dan efektivitas dalam mempelajari algoritma pengurutan. Penelitian ini bertujuan untuk menghasilkan aplikasi permainan yang menjelaskan langkah-langkah algoritma bubble sort dengan melibatkan interaksi pengguna dalam bentuk permainan. Aplikasi dibangun menggunakan bahasa pemrograman Java dan dijalankan pada sistem operasi Android sehingga untuk menjalankan aplikasi hanya memerlukan sebuah perangkat smartphone. Metode penelitian ini dilakukan dengan cara menganalisis teori-teori yang terkait, aplikasi dan penelitian serupa. Kebutuhan yang diperoleh dari hasil analisis digunakan pengembangan aplikasi. Pengembangan aplikasi menggunakan model waterfall yang terdiri dari tahap analisis, desain, pengkodean dan pengujian. Hasil penelitian berupa aplikasi permainan komputer yang dapat digunakan untuk pembelajaran algoritma bubble sort sehingga diharapkan dapat membantu seseorang dalam memahami langkah-langkah pengurutan dengan algoritma bubble sort.*

Kata kunci: *Android, bubble sort, permainan komputer, Java*

I. PENDAHULUAN

Perkembangan teknologi yang pesat banyak dimanfaatkan untuk melakukan banyak hal dalam berbagai bidang. Penerapan teknologi dalam kegiatan sehari-hari dapat memudahkan dan mempercepat pekerjaan terselesaikan. Salah satu bidang yang memanfaatkan teknologi adalah pendidikan. Penerapan teknologi dalam bidang pendidikan dapat mempermudah tugas seorang pengajar dalam memberikan materi sebuah pelajaran. Salah satu cara penerapan teknologi dalam bidang pendidikan adalah dengan menyajikan suatu materi pelajaran dengan cara yang menarik agar dapat mudah dimengerti. Aplikasi permainan merupakan cara yang dapat menarik minat individu dalam belajar. Banyak aplikasi edukasi yang dibuat dalam bentuk permainan sehingga dapat melatih kemampuan untuk mendalami sebuah atau beberapa materi pelajaran.

Algoritma merupakan salah satu materi dasar yang harus dikuasai dalam ilmu pemrograman komputer. Banyak aplikasi yang dibuat menggunakan algoritma untuk membantu menyelesaikan masalah dalam kegiatan sehari-hari. Salah satu algoritma dasar yang diajarkan dalam ilmu pemrograman komputer adalah algoritma pengurutan *(sorting)*.

Pengurutan adalah algoritma yang mempelajari teknik pengurutan data dalam program komputer. Data dapat diurutkan mulai dari yang terkecil hingga terbesar *(ascending)* atau dari yang terbesar hingga terkecil *(descending)*. Dalam ilmu pemrograman komputer, pengurutan sering digunakan untuk mengoptimalkan proses pencarian data. Pengurutan memiliki beberapa metode dalam mengurutkan data namun metode yang paling sederhana adalah *bubble sort*. Berdasarkan latar belakang tersebut maka pada tugas akhir ini penulis membuat sebuah aplikasi permainan edukasi yang menjelaskan langkah-langkah proses pengurutan menggunakan algoritma *bubble sort*. Aplikasi ini diharapkan dapat membantu individu yang sedang mempelajari algoritma *bubble sort* dalam ilmu pemrograman komputer.

Berdasarkan latar belakang yang telah dibahas maka rumusan masalah yang dapat disimpulkan adalah bagaimana merancang dan membangun aplikasi permainan yang dapat membantu seseorang dalam mempelajari algoritma *bubble sort*. Tujuan penelitian dari permasalahan di atas adalah untuk membangun aplikasi yang dapat membantu seseorang yang sedang mempelajari konsep dan langkah-langkah proses algoritma *bubble sort*.

II. METODE PENELITIAN

Metode yang digunakan dalam melakukan penelitian ini adalah: (1). Studi Pustaka. Studi pustaka dilakukan dengan cara mencari informasi dari buku referensi, jurnal, sumber-sumber program dari internet yang berkaitan dengan materi algoritma *bubble sort*, pengembangan *game Android*, dan bahasa pemrograman *Java*; dan (2) Metode Pengembangan Peranti Lunak. Pengembangan sistem aplikasi permainan untuk pembelajaran algoritma *bubble sort* menggunakan *waterfall SDLC model* yang terdiri dari tahapan analisis, desain, implementasi, pengujian dan pemeliharaan aplikasi.

A. Algoritma

Algoritma adalah urutan langkah-langkah proses penyelesaian masalah secara logika yang disusun dalam bentuk tertulis agar dapat mudah dimengerti. Dalam kehidupan sehari-hari, banyak kegiatan yang dapat diselesaikan dengan algoritma. Cara membuat masakan yang dinyatakan dalam resep berisi langkah-langkah membuat masakan dapat disebut sebagai algoritma.

Dalam ilmu komputer, algoritma merupakan materi dasar yang harus dikuasai sebelum mempelajari bahasa pemrograman. Algoritma digunakan untuk menyelesaikan berbagai permasalahan secara terstruktur dalam pemrograman komputer. Definisi algoritma menurut beberapa sumber antara lain: (1) M. Zarlis dan Handrizal mendefiniskan algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis dan logis [1]; (2) Rosa A.S. dan M. Shalahuddin menyatakan dalam dunia pemrograman, algoritma adalah solusi dari suatu masalah yang harus dipecahkan dengan menggunakan komputer [2]; dan (3) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest dan Clifford Stein menyatakan bahwa algoritma adalah prosedur komputasi yang mengambil beberapa nilai sebagai masukan *(input)* dan menghasilkan nilai sebagai keluaran *(output)*. Algoritma adalah urutan langkah-langkah yang memproses masukan sehingga menghasilkan keluaran [3].

Untuk menghasilkan keluaran yang baik dan sesuai kriteria aplikasi yang diharapkan maka algoritma yang diterapkan dalam program harus benar. Kelebihan algoritma menurut M. Zarlis dan Handrizal (2008:3) di antaranya: Algoritma tidak terikat dengan bahasa pemrograman manapun, penulisan algoritma independen dari bahasa pemrograman. Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman. Apapun bahasa pemrogramannya, keluaran yang dihasilkan akan sama jika algoritmanya juga sama [1]. Contoh algoritma dalam menghitung keliling persegi panjang dengan komputer adalah: (1) Dapatkan nilai panjang persegi panjang; (2) Dapatkan nilai lebar persegi panjang; (3) Hitung keliling persegi panjang dengan rumus 2 x (panjang + lebar); dan (4) Cetak hasil keliling persegi panjang. Algoritma di atas dinyatakan dalam bahasa manusia yang mudah dimengerti. Dalam ilmu komputer, algoritma dapat dinyatakan dalam dua bentuk algoritma yaitu diagram alir (*flowchart*) dan *pseudocode*.

B. Sorting

Sorting merupakan proses pengurutan data berdasarkan aturan yang dikehendaki. Pengurutan data merupakan proses yang sangat dibutuhkan dalam pemrograman [4]. Data yang terurut mudah untuk dicari, mudah diperiksa dan diperbaiki jika terdapat kesalahan. Pengurutan dapat dilakukan mulai dari data terkecil hingga data terbesar *(ascending*) atau dari data terbesar hingga data terkecil *(descending)*. Ada banyak metode yang digunakan dalam melakukan pengurutan di antaranya adalah *insertion sort, selection sort*, dan *bubble sort*. Metode yang

paling sederhana dan mudah dipelajari dalam sorting adalah *bubble sort* [5].

C. Insertion Sort

Metode *insertion sort* bekerja seperti orang yang mengurutkan kartu di tangan. Dimulai dengan tangan kiri yang kosong dan kartu yang tertumpuk di meja. Selanjutnya kartu di meja diambil satu persatu dan diletakkan di tangan kiri dengan posisi yang terurut. Untuk menemukan posisi yang benar, maka kita harus membandingkan kartu yang ada di tangan kiri secara berurutan dari kanan ke kiri [3].

D. Selection Sort

Prinsip dari metode *selection sort* adalah untuk mencari data terkecil dengan membandingkan angka pertama sampai dengan angka terakhir, kemudian angka terkecil tersebut ditukar dengan angka pertama. Proses selanjutnya mencari data terkecil dengan membandingkan angka kedua sampai dengan angka terakhir, kemudian angka terkecil yang diperoleh ditukar dengan angka kedua [6].

E. Bubble Sort

Bubble sort mengurutkan angka dengan cara membandingkan dua angka yang bersebelahan. Jika data diurutkan secara *ascending* maka komputer akan membandingkan angka pertama dengan angka berikutnya, jika angka pertama lebih besar maka kedua angka pembanding akan bertukar posisi sehingga angka yang lebih besar akan ditempatkan di sebelah kanan, proses membandingkan angka akan terus dilakukan mulai dari angka pertama hingga angka terakhir dalam suatu deret angka hingga diperoleh angka terbesar dalam 1 iterasi. Setelah angka terbesar diperoleh maka akan ditempatkan pada angka terakhir dalam suatu deret angka. Proses perbandingan dan pertukaran selanjutnya untuk mencari angka terbesar kedua untuk ditempatkan pada angka di sebelah kiri angka terbesar yang telah diperoleh sebelumnya. Proses pengurutan akan terus dilakukan sampai semua data terurut. Tahap-tahap dalam proses pengurutan algoritma *bubble sort* mengurutkan data secara *ascending* dapat dijelaskan dengan pada gambar-gambar berikut.

9	5	10	3	7	8

Gambar 1. Posisi awal angka

Pada proses pertama, angka pertama akan dibandingkan dengan angka ke-2. Jika angka pertama lebih besar dari angka ke-2 maka isi kedua angka akan ditukar. Posisi kedua angka setelah ditukar pada gambar 2.

5	9	10	3	7	8

Gambar 2. Pengurutan angka dengan bubble sort

Berikutnya angka ke-2 akan dibandingkan dengan angka ke-3. Jika isi angka ke-2 lebih kecil maka kedua angka tidak akan ditukar. Posisi angka setelah perbandingan angka ke-2 dan ke-3 pada gambar 3.

5	9	10	3	7	8

Gambar 3. Pengurutan angka dengan bubble sort

Selanjutnya perbandingan dilakukan terhadap angka ke-3 dan ke-4. Karena angka ke-3 lebih besar dari angka ke-4 maka kedua isi angka tersebut akan bertukar posisi. Gambar 4 menggambarkan posisi kedua angka setelah ditukar:

5	9	3	10	8	7

Gambar 4. Pengurutan angka pada bubble sort

Perbandingan dilanjutkan terhadap angka ke-4 dan angka ke-5. Angka ke-4 lebih besar dari angka ke-5 sehingga isi kedua angka ditukar. Hasil setelah pertukaran dapat dilihat pada gambar 5.

5	9	3	7	10	8

Gambar 5. Pengurutan angka dengan bubble sort

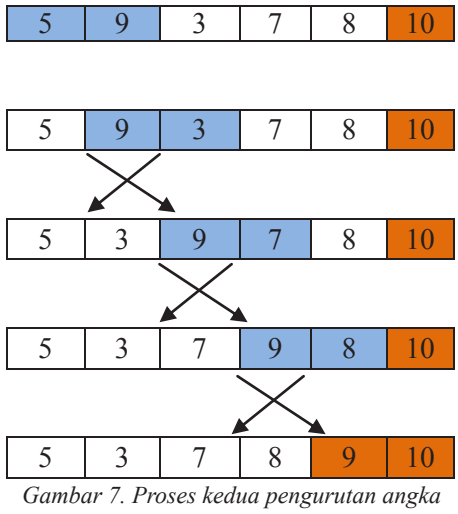
Berikutnya angka ke-5 akan dibandingkan dengan angka ke-6 sehingga didapatkan bilangan terbesar pertama dan ditempatkan pada indeks terakhir. Hasil perbandingan seperti pada gambar 6.

5	9	3	7	8	10

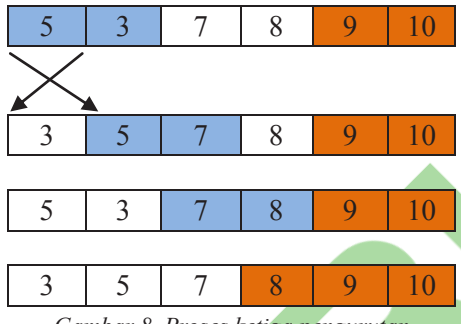
Gambar 6. Pengurutan angka dengan bubble sort

Setelah angka terbesar pertama diperoleh maka pengurutan dilanjutkan dengan mencari angka terbesar kedua. Perbandingan dilakukan dari elemen ke-1 hingga elemen ke-5 karena elemen ke-6 sudah berisi angka terbesar. Gambar 7 menjelaskan proses pengurutan sehingga diperoleh angka terbesar kedua.

Setelah angka terbesar kedua diperoleh maka pengurutan angka dilanjutkan dengan mencari angka terbesar selanjutnya hingga semua angka berurutan. Tahap mencari angka terbesar berikutnya dijelaskan pada gambar 8, gambar 9 dan gambar 10.

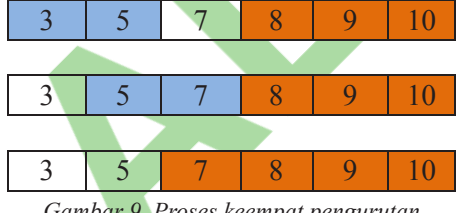


Gambar 7. Proses kedua pengurutan angka dengan bubble sort



Gambar 8. Proses ketiga pengurutan dengan bubble sort

Gambar 9 menjelaskan proses perbandingan angka untuk mencari angka terbesar berikutnya.

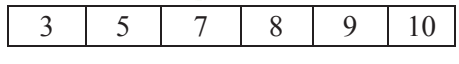


Gambar 9. Proses keempat pengurutan dengan bubble sort

Pada gambar 10 proses perbandingan dilakukan pada 2 angka yang tersisa.

Gambar 10. proses kelima pengurutan dengan bubble sort

Setelah melalui proses kelima pengurutan angka dengan metode *bubble sort* maka hasil akhir angka yang telah terurut seperti pada gambar 11.



Gambar 11. Posisi angka setelah pengurutan dengan bubble sort

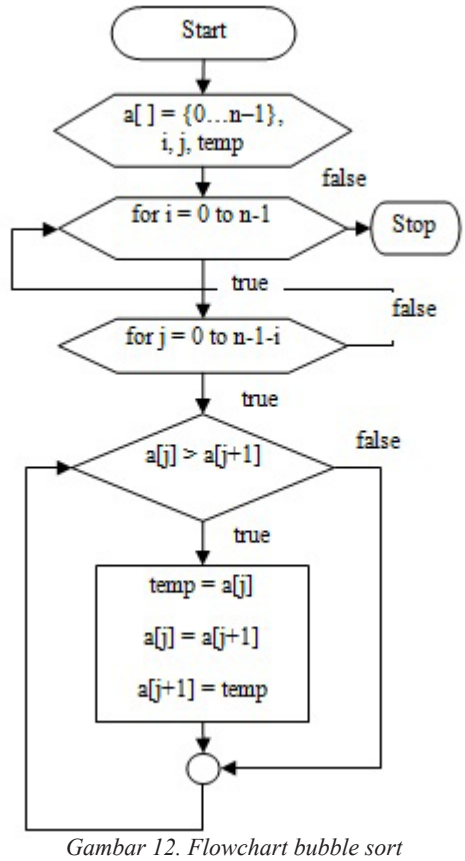
Dari proses pengurutan yang telah dilakukan, maka langkah-langkah algoritma *bubble sort* dapat ditulis:

1. Bandingkan posisi data ke I = 0 dan J = 1, Jika data di posisi I lebih besar dari pada data di posisi J, maka tukar posisi kedua data.
2. Variabel I ditambah 1.
3. Variabel J ditambah 1.
4. Ulangi tahap ke-1, 2, 3 hingga variabel I = N-2 dan variabel J = N-1.
5. Untuk tahap selanjutnya data yang dibandingkan akan berkurang karena data terbesar di letakkan pada sebelah kanan data.

Berdasarkan langkah-langkah yang telah diuraikan, maka algoritma *bubble sort* dapat ditulis dalam *pseudocode*. Penulisan algoritma *bubble sort* dalam *pseudocode* seperti yang telah diadaptasi dan dimodifikasi untuk keperluan penelitian [7].

```
BubbleSort( int a[], int n)
begin
 for i = 0 to n-1
  for j = 0 to n-1-i
   if a[j] > a[j+1]
     temp = a[j]
     a[j] = a[j+1]
     a[j+1] = temp
   end if
  end for
 end for
end
```

Algoritma *bubble sort* dapat ditulis dalam bentuk *flowchart* , lihat gambar 12 yang telah diadaptasi dari *internet* [8] dan diubah untuk keperluan penelitian:



Gambar 12. Flowchart bubble sort

E.Belajar

Definisi belajar menurut beberapa sumber di antaranya, menurut Thursan Hakim, belajar adalah suatu proses perubahan di dalam kepribadian manusia, dan perubahan tersebut ditampakkan dalam bentuk peningkatan kualitas dan kuantitas tingkah laku seperti peningkatan kecakapan, pengetahuan, sikap, kebiasaan, pemahaman, ketrampilan, daya pikir dan kemampuan lain-lain [9], sedangkan menurut Cronbach seperti yang dikutip oleh Agus Suprijono, belajar adalah perubahan perilaku sebagai hasil dari pengalaman [10], selain itu pernyataan Morgan melalui Agus Suprijono mendefinisikan belajar adalah perubahan perilaku yang bersifat permanen sebagai hasil dari pengalaman [10]. Dari pengertian di atas dapat disimpulkan bahwa belajar merupakan kegiatan yang dilakukan sepanjang hidup manusia. Aktivitas belajar dapat membawa perubahan terhadap seseorang, baik perubahan pengetahuan, sikap dan keterampilan.

F. Model Pembelajaran

Salah satu model pembelajaran yang dapat digunakan untuk mengukur proses mempelajari algoritma berdasarkan obyektif yang dituju adalah Taksonomi *Bloom*. Dalam taksonomi Bloom ada enam obyektif yang dituju yang dibedakan dalam enam tingkatan yang dijelaskan sebagai berikut [11]: (1) Proses pembelajaran dimulai dengan proses melihat, peserta didik melihat teks penjelasan algoritma untuk mendapatkan pengetahuan; (2) Kemudian peserta didik melihat *flowchart* algoritma dan demo permainan untuk memahami proses dalam algoritma; (3) Berikutnya dalam tahap bermain, peserta didik menerapkan pengetahuan tentang algoritma untuk bermain *game* algoritma; (4) Permainan dibagi dalam beberapa langkah, tiap langkah mensimulasikan satu langkah algoritma. Selama bermain, peserta didik dapat menganalisis langkah-langkah dalam algoritma kemudian menghubungkan setiap langkah untuk memenangkan permainan; (5) Peserta didik dapat mengevaluasi kecepatan, kompleksitas, dan efisiensi dari suatu algoritma dengan memainkan permainan yang terkait. Misalnya jika permainan sulit dimenangkan, maka dapat berarti algoritma yang diterapkan terlalu kompleks; dan (6) Sebagai hasil dari pembelajaran, peserta didik dapat menganalisis algoritma dan *game* yang dimainkan untuk mendesain permainan algoritma yang baru. Kemudian peserta didik dapat mengevaluasi hasil desain permainan algoritma yang baru.

G. Belajar Algoritma

Algoritma membantu mahasiswa mengembangkan daya penalaran atau kerangka berpikir yang sistematis dalam memahami masalah dan membuat perencanaan atau konsep pemecahan masalah yang lebih baik sehingga dapat menghasilkan solusi pemecahan masalah yang tepat [12]. Dengan mempelajari algoritma, mahasiswa dapat terlatih untuk berpikir secara terstruktur dalam menyelesaikan masalah.

H. Mobile Learning

Ariesto Hadi Sutopo menyatakan *Mobile Learning* atau *m-learning* adalah pembelajaran dengan menggunakan alat bantu perangkat bersifat *mobile* seperti *PDAs*, *mobile phones, laptop* dan peralatan teknologi lain untuk pembelajaran [13]. Perkembangan teknologi informasi dan komunikasi yang semakin pesat memungkinkan pembelajaran dilakukan dengan perangkat *mobile*. Keuntungan menggunakan *m-learning* menurut Ariesto Hadi Sutopo [13] adalah: (1) *Convenience*, pengguna dapat mengakses dari mana saja pada konten pembelajaran termasuk kuis, jurnal, *game*, dan lainnya; (2) *Collaboration*, pembelajaran dapat segera dilakukan setiap saat secara *real time*; (3) *Portability*, penggunaan buku diganti dengan RAM dengan pembelajaran yang dapat diatur dan dihubungkan; (4) *Compatibility*, pembelajaran dirancang untuk digunakan pada perangkat *mobile*; dan (5) *Interesting*, pembelajaran yang dikombinasikan dengan *game* akan menyenangkan.

Berdasarkan poin di atas maka dapat ditarik kesimpulan bahwa pelajaran yang dikemas dalam bentuk *mobile game* dapat menarik minat individu dalam belajar.

I. Permainan

Kata permainan yang jika diterjemahkan ke dalam Bahasa Inggris yaitu *game*. Menurut Joan Freeman dan Utami Munandar seperti dikutip oleh Andang Ismail, *game* adalah suatu aktivitas yang membantu anak mencapai perkembangan yang utuh, baik fisik, intelektual, sosial, moral, dan emosional [14]. Menurut Michael Zyda *game* adalah ajang untuk melatih fisik atau mental dengan tujuan menghibur atau untuk mendapatkan penghargaan [15].

J. Permainan Komputer

Menurut Jouni Smed dan Harri Hakonen, permainan komputer adalah permainan yang

dilakukan dengan bantuan program komputer [16], sedangkan menurut kamus online *Cambridge,* permainan komputer adalah permainan yang dimainkan di komputer, di mana gambar yang muncul di layar dikendalikan dengan menekan tombol atau menggerakkan *joystick* [17].

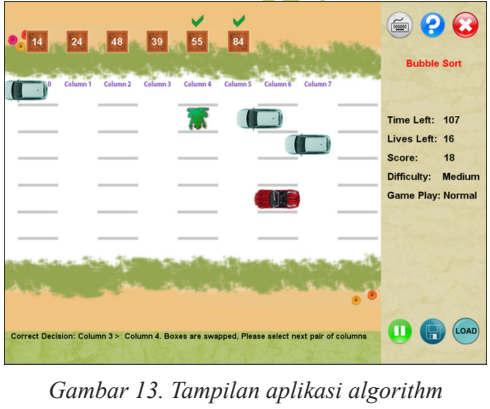
Sahar Sabanah dan Dr. Jim X. Chen menyatakan pembelajaran algoritma dengan menggunakan permainan komputer dipilih karena memiliki beberapa nilai kelebihan [11]: (1) Permainan komputer telah popular, berbagai individu dari kategori umur remaja dan dewasa di seluruh dunia telah mengenal permainan komputer; (2) Permainan komputer lebih interaktif, permainan yang interaktif dapat mendorong pengguna untuk berpikir dan bertindak; (3) Permainan komputer lebih kompetitif, permainan komputer dapat meningkatkan motivasi pengguna; (4) Permainan komputer mempermudah penilaian, pemahaman seseorang terhadap algoritma dapat dinilai dari keberhasilan dalam menyelesaikan permainan; dan (5) Permainan komputer menyediakan unsur hiburan, pembelajaran algoritma dapat lebih menarik dan menyenangkan dengan permainan komputer.

K. Aplikasi Pembelajaran *Bubble Sort*

Pada saat tulisan ini dibuat, sudah ada beberapa aplikasi yang dibuat untuk pembelajaran algoritma *bubble sort*, beberapa di antaranya adalah:

1.Sorting Algortihm 2D Game

Sorting Algorithm 2D Game menggunakan objek katak untuk melakukan pengurutan. Untuk menyelesaikan permainan, pengguna harus mengurutkan kotak yang berisi angka acak sesuai dengan aturan algoritma *bubble sort*. Pada aplikasi ini telah memiliki beberapa fitur yang sesuai dengan Taksonomi *Bloom,* diantaranya penulisan algoritma *bubble sort* dalam *pseudocode* dan *flowchart,* serta tutorial cara menjalankan permainan. Namun

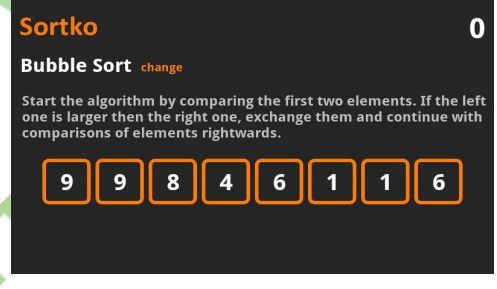


Gambar 13. Tampilan aplikasi algorithm 2D game

dibutuhkan waktu yang lebih lama untuk mengurutkan deretan angka yang sedikit. Aplikasi hanya dijalankan pada perangkat berbasis *Windows* dan dapat diunduh di alamat *website*: *http://sortingalgorithmsgame.webs.com/*

2.Sortko

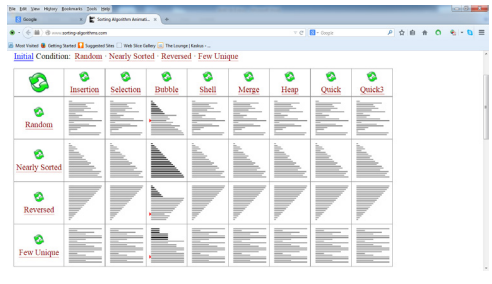
Sortko adalah aplikasi pembelajaran algoritma pengurutan yang dijalankan pada perangkat *mobile* yang berbasis *android*. Aplikasi Sortko menyediakan beberapa metode pengurutan yang dapat dipilih oleh pemain (*Bubble Sort, Insertion Sort, Quick Sort, Shell Sort, Selection Sort*). Aplikasi hanya terdiri dari 3 tampilan yaitu, tampilan awal, tampilan sorting, dan tampilan hasil. Dalam proses pengurutan, aplikasi menyediakan teks panduan yang dapat membantu pengguna dalam menyelesaikan pengurutan. Namun aplikasi tidak memiliki tutorial pada awal *game* dijalankan seperti ketentuan pada taksonomi Bloom. Setelah proses pengurutan selesai maka hasil pengurutan dapat dilihat pada tampilan hasil. Tampilan hasil juga berisi daftar 50 nilai tertinggi dari seluruh pengguna aplikasi. Aplikasi dapat diunduh di *Google Play Store*.



Gambar 14. Tampilan bubble sort pada sortko

3.Sorting Algorithm Animation

Sorting Algorithm Animation merupakan aplikasi visualisasi algoritma *sorting* yang berbasis *web* dan dijalankan pada *web browser*. Pengguna dapat membandingkan operasi pengurutan dalam beberapa metode yang berbeda secara bersamaan. Aplikasi masih memiliki kekurangan karena pengguna tidak dapat melakukan pengurutan secara

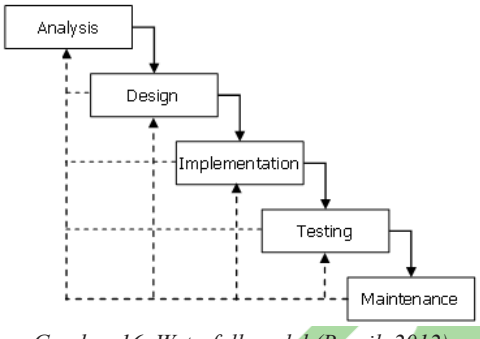


Gambar 15. Tampilan aplikasi sorting algorithm animation

mandiri. Aplikasi dapat dijalankan di alamat *website: http://www.sorting-algorithms.com/*

L. Waterfall SDLC Model

Waterfall SDLC Model adalah metode pengembangan aplikasi yang bersifat sekuensial dimana tahap-tahap pengembangan aplikasi disusun secara berurutan menurun seperti air terjun (*waterfall*) [18]. Pengerjaan tahap-tahap dalam *waterfall model* dilakukan secara berurutan. Jika tahap pertama belum selesai dikerjakan maka tidak bisa melanjutkan ke tahap kedua, ketiga dan seterusnya. Tahap ketiga bisa dilaksanakan jika tahap pertama dan kedua telah selesai dikerjakan.

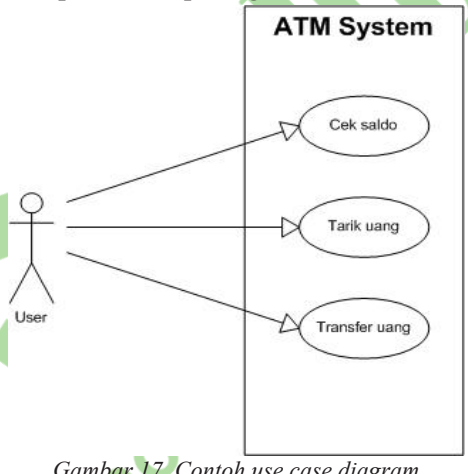


Gambar 16. Waterfall model (Bassil, 2012)

Tahap-tahap pengembangan aplikasi dalam metode *waterfall SDLC Model* meliputi: (1) Analisis. Tahap analisis dalam penelitian ini meliputi analisa terhadap kebutuhan aplikasi. Pengumpulan data yang terkait penelitian melalui studi pustaka dan analisis *use case* terhadap apa yang dapat dilakukan sistem aplikasi sebagai respon kepada pengguna. Pada tahap ini juga akan dibuat *activity diagram, sequence diagram, statechart diagram* terhadap sistem aplikasi yang sedang dibangun; (2) Desain. Tahap desain akan menterjemahkan hasil analisa kebutuhan aplikasi yang diperoleh dari tahap analisa ke dalam arsitektur perangkat lunak dengan menggunakan *class diagram*. Pada tahap ini juga termasuk desain *user interface* dari aplikasi; (3) Implementasi. Tahap implementasi merupakan tahap pembuatan program dari hasil analisa dan desain yang telah dilakukan pada tahap sebelumnya. Setiap modul program yang telah dibuat akan diuji untuk menemukan kesalahan dan kemudian bisa diperbaiki; (4) Pengujian. Setelah pembuatan program selesai dikerjakan maka pengujian dilakukan terhadap keseluruhan aplikasi untuk memastikan aplikasi telah sesuai dengan analisa kebutuhan aplikasi pada tahap analisa dan spesifikasi aplikasi pada tahap desain; dan (5) Pemeliharaan. Pemeliharaan dan pengembangan aplikasi diperlukan ketika adanya perubahan atau penambahan fitur terhadap aplikasi yang sudah berjalan.

M. Use Case Diagram

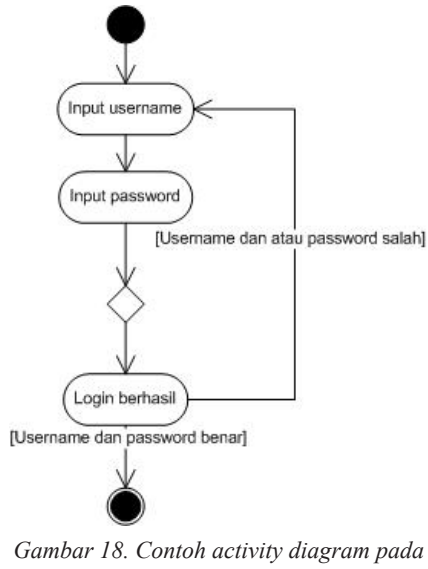
Untuk melakukan tahap analisis pada perancangan sebuah sistem (peranti lunak) terdapat tahap penggambaran fungsionalitas sistem. Untuk menggambarkan fungsionalitas sebuah sistem (peranti lunak) digunakan *use case diagram.* Sebuah *use case diagram*, merepresentasikan sebuah interaksi antara aktor dengan sistem [19]. Aktor dalam *use case* adalah bagian luar dari sistem yang berinteraksi dengan sistem. Aktor dapat berupa manusia yang menggunakan sistem atau mesin yang berinteraksi dengan sistem. Sedangkan sistem itu sendiri dapat berupa aplikasi peranti lunak. Contoh *use case diagram* dapat dilihat pada gambar 17.



Gambar 17. Contoh use case diagram dalam sistem ATM

N. Activity Diagram

Untuk menggambarkan berbagai aktivitas yang terjadi dalam sistem maka digunakan *activity diagram*. *Activity diagram* menggambarkan alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin

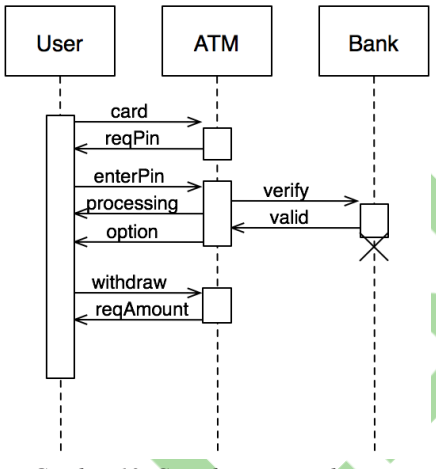


Gambar 18. Contoh activity diagram pada proses login

terjadi, dan bagaimana sebuah aktivitas berakhir [19]. *Activity diagram* menggambarkan aktivitas-aktivitas yang terjadi dalam satu *use case* atau lebih. Contoh *activity diagram* dapat dilihat pada gambar 18.

O. Sequence Diagram

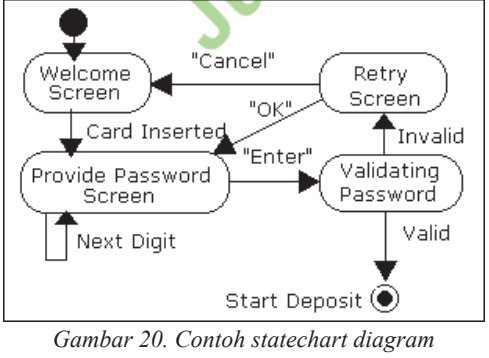
Untuk menggambarkan interaksi antar objek dalam sistem dalam urutan waktu tertentu maka digunakan *sequence diagram*. *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) [19]. Interaksi dalam *sequence diagram* berupa pesan yang dikirimkan antar objek dalam urutan waktu tertentu. Contoh *sequence diagram* dapat dilihat pada gambar 19.



Gambar 19. Contoh sequence diagram dalam sistem ATM

P. Statechart Diagram

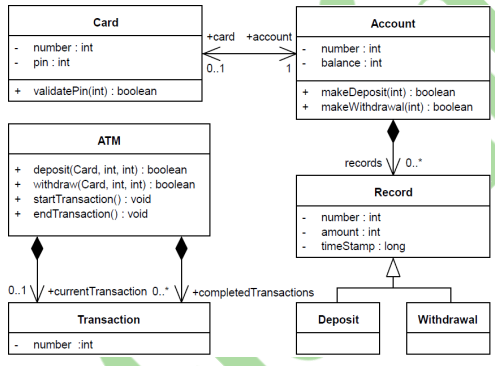
Untuk menggambarkan perubahan keadaan dalam sistem terhadap proses aktivitas dalam peranti lunak maka digunakan *statechart diagram*. *Statechart diagram* menggambarkan transisi perubahan keadaan (*state*) dari satu keadaan ke keadaan lainnya pada peranti lunak sebagai akibat dari stimuli yang diterima [19]. *Statechart diagram* menunjukkan keadaan objek dan proses yang menyebabkan perubahan keadaan dalam peranti lunak. Contoh *statechart diagram* dapat dilihat pada gambar 20.



Gambar 20. Contoh statechart diagram pada sistem ATM

Q. Class Diagram

Untuk menggambarkan struktur dan hubungan antar *class* dalam sistem maka digunakan *class diagram*. *Class diagram* menggambarkan struktur dan deskripsi *class* beserta hubungan *class* satu sama lain [19]. *Class diagram* memberikan pandangan dari suatu sistem dengan menunjukkan *class* yang ada dalam sistem dan hubungan antar *class*. Contoh *class diagram* dapat dilihat pada gambar 21.



Gambar 21. Contoh class diagram pada sistem bank

III. HASIL DAN PEMBAHASAN

A. Analisis

Analisis dilakukan untuk mengetahui apa saja yang dibutuhkan sebuah aplikasi yang dapat menjelaskan kepada pengguna tentang definisi algoritma *bubble sort* dan langkah-langkah pengurutan angka dengan metode *bubble sort* serta unsur apa saja yang dibutuhkan aplikasi dari sisi permainan.

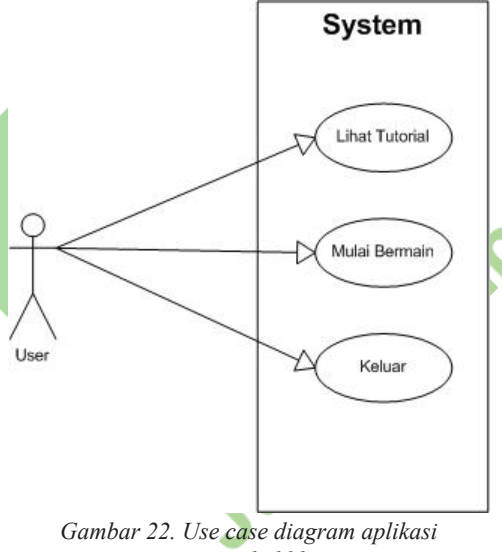
- ***Analisis Kebutuhan Aplikasi***

Berdasarkan kajian pustaka dan kajian aplikasi yang sudah ada, aplikasi yang akan dibangun harus memiliki fitur sebagai berikut: (1) Aplikasi memiliki penjelasan singkat tentang algoritma *bubble sort*. Pada saat aplikasi dijalankan, pengguna dapat membaca pengertian dan *flowchart* algoritma *bubble sort*; (2) Aplikasi memiliki tutorial cara menjalankan aplikasi. Tutorial dapat dijalankan sebelum memulai permainan sehingga pengguna dapat mempelajari cara bermain dan konsep pengurutan dengan metode *bubble sort;(3)* Aplikasi memiliki teks bantuan yang dapat membantu pengguna dalam menyelesaikan permainan. Teks bantuan dapat membantu pengguna dalam mengantisipasi kesalahan langkah yang dilakukan oleh pengguna dan membantu pengguna untuk menyelesaikan permainan; (4) Pengguna dapat menganalisa setiap langkah yang dilakukan untuk menyelesaikan permainan. Aplikasi

permainan memiliki panduan tahap demi tahap untuk menyelesaikan permainan; (5) Pengguna dapat menghubungkan langkah penyelesaian permainan untuk diaplikasikan kepada permainan sejenis dengan masalah berbeda. Aplikasi yang dibangun dapat membantu pengguna dalam mengeksplorasi dan bereksperimen terhadap algoritma *bubble sort* sehingga pengguna dapat mengerti dan menyelesaikan aplikasi permainan lain yang serupa; (6) Pengguna dapat mengevaluasi peningkatan kemampuan dalam menyelesaikan permainan algoritma *bubble sort*. Pengguna dapat mengevaluasi kemampuan dalam memahami algoritma *bubble sort* setelah menyelesaikan setiap level dalam permainan; dan (7) Aplikasi permainan memiliki unsur interaktif. Pengguna dapat berinteraksi untuk menyelesaikan permainan. Aplikasi permainan interaktif dapat mendorong pengguna untuk berpikir dan bertindak dalam menyelesaikan permainan.

- ***Analisis Use Case***

Use case diagram menggambarkan interaksi antara pengguna dengan sistem dan fungsi-fungsi apa saja yang dapat dilakukan sebuah sistem terhadap respon pengguna. Pada aplikasi pengurutan dengan *bubble sort* memiliki 3 *use case* yaitu lihat tutorial, mulai bermain, dan keluar. *Use Case diagram* aplikasi permainan *bubble sort* dapat dilihat pada gambar 22.



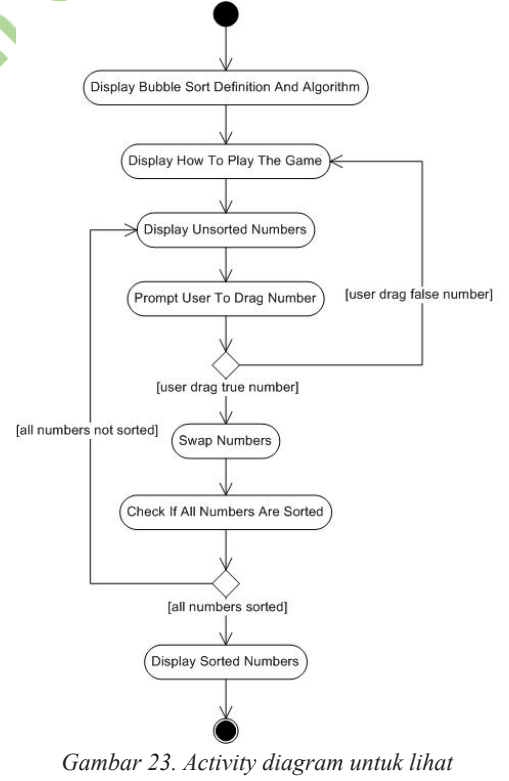
Gambar 22. Use case diagram aplikasi permainan bubble sort

- ***Analisis Activity Diagram***

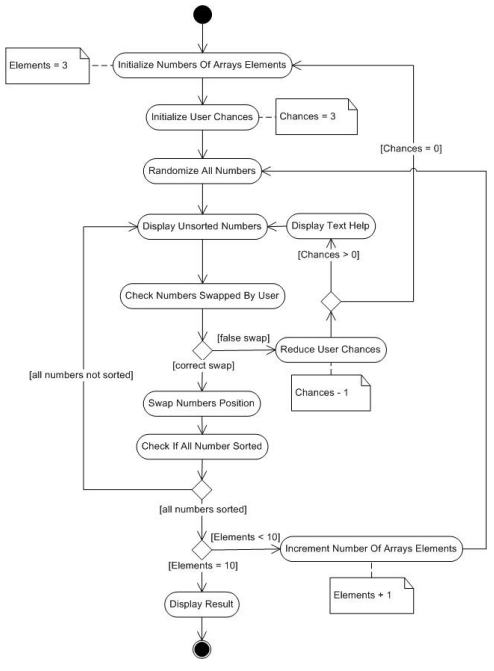
Activity diagram menggambarkan alur aktivitas masing-masing *use case* dalam sistem yang sedang dirancang mulai dari awal hingga berakhirnya aktivitas. Berdasarkan *use case diagram* pada gambar 17, maka dapat digambarkan dua *activity diagram* yaitu *activity diagram* untuk Lihat Tutorial dan *activity diagram* untuk Mulai Bermain.

Activity diagram untuk Lihat Tutorial menampilkan teks penjelasan mengenai definisi pengurutan angka dengan metode *bubble sort* dan menampilkan *pseudocode* algoritma *bubble sort*. Setelah tampilan teks penjelasan definisi *bubble sort* selesai maka pengguna akan diarahkan ke tampilan tutorial cara mengurutkan angka dengan metode *bubble sort.* Aplikasi akan menampilkan deretan angka yang sudah teracak dan pengguna akan dibimbing untuk mengurutkan deretan angka sampai semua angka sudah terurut. *Activity diagram* untuk Lihat Tutorial dapat dilihat pada gambar 23.

Activity diagram untuk Mulai Bermain pada gambar 24 menggambarkan proses pengurutan angka oleh pengguna dimulai dari deretan 3 angka yang sudah diacak kemudian pengguna dapat mengurutkan angka yang tersedia hingga semua angka terurut. Setelah semua angka terurut, maka permainan akan dilanjutkan pada level berikutnya. Aplikasi akan kembali menampilkan level selanjutnya berupa deretan angka yang telah teracak. Deretan angka yang tersedia akan bertambah satu menjadi 4 deretan angka acak, dan begitu seterusnya hingga deretan angka mencapai sepuluh deretan angka acak. Pengguna hanya memiliki 3 kesempatan untuk melakukan kesempatan, jika pengguna melakukan lebih dari 3 kali kesalahan pengurutan maka permainan akan di-*reset* mulai dari awal dan pengguna harus menyelesaikan permainan dari level pertama.



Gambar 23. Activity diagram untuk lihat tutorial



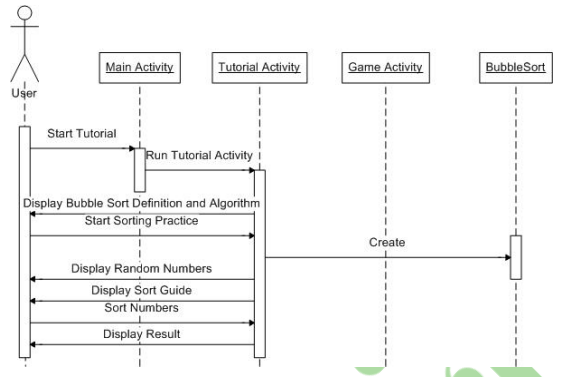
Gambar 24. Activity diagram untuk mulai bermain

- ***Analisis Sequence Diagram***

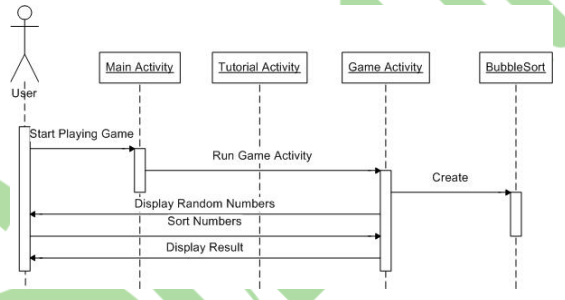
Sequence diagram menggambarkan interaksi antara pengguna dengan objek-objek di dalam sistem aplikasi yang sedang dirancang secara berurutan menurut waktu kejadian dalam suatu *use case*. Dalam *sequences diagram* aplikasi pengurutan *bubble sort* terdapat 4 buah objek yaitu *main activity, tutorial activity, game activity,* dan *bubble sort*.

Sequence diagram pada gambar 25 menggambarkan skenario pengguna mengobservasi tutorial aplikasi pengurutan dengan metode *bubble sort.* Skenario Lihat Tutorial dimulai dari pengguna yang memilih menjalankan tutorial pada *main activity* yang kemudian akan memanggil *tutorial activity*. Pada saat dijalankan, *tutorial activity* akan menampilkan definisi dan algoritma *bubble sort* yang dapat dibaca oleh pengguna sebelum mulai bermain mengurutkan angka. Setelah melihat definisi dan algoritma *bubble sort,* pengguna dapat memulai tutorial mengurutkan beberapa angka yang sudah teracak dan hasil pengurutan akan ditampilkan.

Gambar 26 menggambarkan *sequence diagram* urutan kejadian pengguna memulai aktivitas bermain mengurutkan angka dengan metode *bubble sort.* Skenario Mulai Bermain dimulai dari pengguna yang memilih memulai permainan pada *main activity* yang kemudian akan memanggil *game activity.* Pada saat dijalankan *game activity* akan menampilkan beberapa angka acak. Pengguna dapat mengurutkan beberapa angka hingga semua angka terurut.



Gambar 25. Sequence diagram untuk lihat tutorial



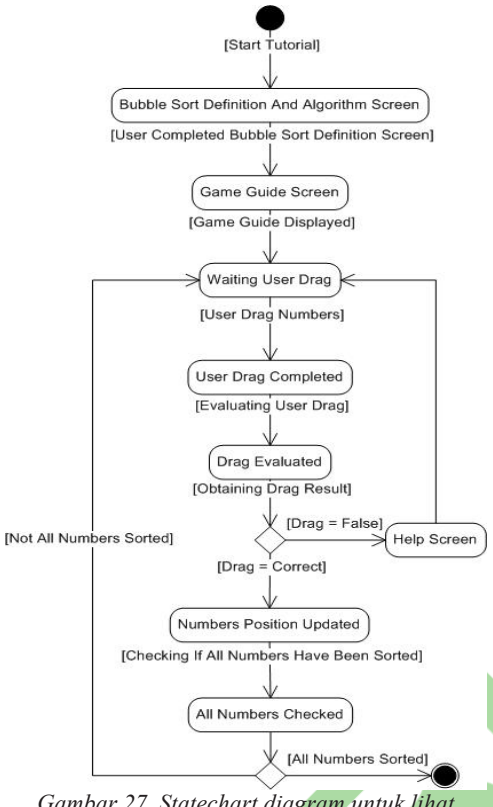
Gambar 26. Sequence diagram untuk mulai bermain

- ***Analisis Statechart Diagram***

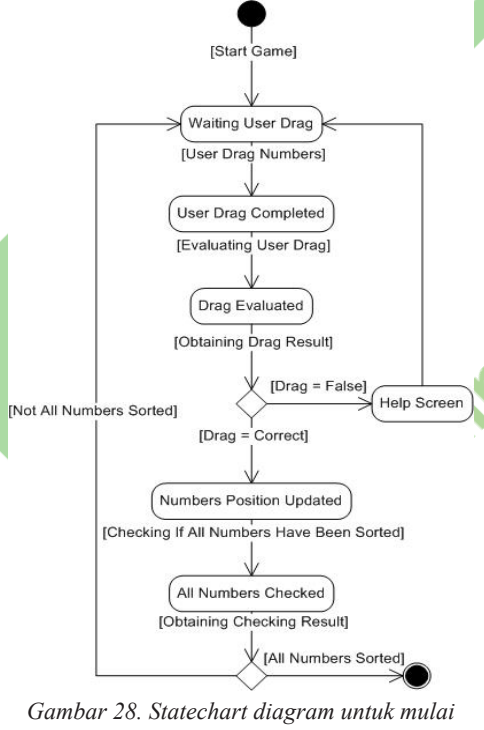
Statechart diagram menggambarkan transisi *(transition)* dan perubahan keadaan *(state)* suatu objek pada sistem dari saat dibuat *(initialize)* hingga objek tersebut dihapus *(destroy)*. *Statechart diagram* dimulai dengan tanda lingkaran hitam solid dan diakhiri dengan tanda lingkaran solid di dalam lingkaran kosong. *State* digambarkan dengan bentuk segiempat dengan sudut membulat dan memiliki keterangan kondisi di dalamnya, sedangkan *transition* digambarkan dengan garis tanda panah dengan keterangan kejadian *(event)* yang memicu perubahan *state*. Gambar 27 menjelaskan urutan perubahan *state* dalam aplikasi pada saat pengguna menjalankan fungsi tutorial aplikasi. Pada saat tutorial dijalankan, aplikasi akan menampilkan penjelasan mengenai definisi dan algoritma *bubble sort* di layar yang dapat dibaca pengguna sebelum memasuki tahapan berikutnya yaitu panduan menjalankan permainan *bubble sort*. Aplikasi akan menampilkan contoh angka acak dan memandu pengguna untuk mengurutkan deretan angka acak dengan metode *bubble sort*.

Statechart diagram pada gambar 28 menggambarkan urutan perubahan *state* pada saat pengguna memulai permainan mengurut deretan angka acak. Aplikasi menampilkan deretan angka acak dan pengguna dapat langsung memulai mengurutkan angka yang tersedia. Aplikasi akan

menampilkan petunjuk benar atau salah dari setiap langkah pengurutan yang dilakukan oleh pengguna.



Gambar 27. Statechart diagram untuk lihat tutorial



Gambar 28. Statechart diagram untuk mulai bermain

2. Desain

- ***Desain Class Diagram***

Class diagram digunakan untuk menggambarkan struktur kelas dalam sistem aplikasi permainan *bubble sort*. *Class diagram* dapat memberikan gambaran interaksi hubungan antar kelas, atribut dan operasi yang dimiliki oleh *class* dalam sistem yang sedang dirancang.

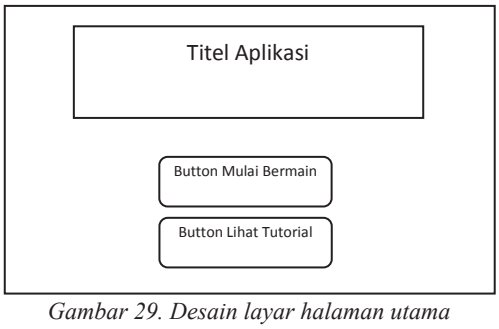
Dalam *class diagram,* notasi kelas digambarkan dengan bentuk kotak yang terdiri dari 3 bagian. Bagian atas dari notasi kelas menjelaskan nama kelas. Bagian tengah notasi kelas menjelaskan atribut yang dimiliki oleh kelas. Bagian bawah notasi kelas menjelaskan operasi yang dimiliki oleh kelas. Tanda garis dan panah menunjukkan hubungan antar kelas.

Class diagram aplikasi permainan *bubble sort* terdiri dari 4 kelas yang berhubungan yaitu kelas *MainActivity, TutorialActivity, GameActivity,* dan *BubbleSort*. Kelas *MainActivity* tidak memiliki atribut dan hanya berisi operasi yang berfungsi untuk menampilkan layar halaman utama aplikasi dan operasi untuk memanggil kelas *TutorialActivity* atau kelas GameActivity sesuai dengan pilihan pengguna.

Kelas *TutorialActivity* berfungsi untuk menampilkan layar tutorial dan menjalankan aktivitas Lihat Tutorial. Kelas *TutorialActivity* memiliki atribut dan operasi yang diperlukan untuk menjalankan tutorial aplikasi pembelajaran *bubble sort*. pada saat awal dieksekusi kelas *TutorialActivity* akan menampilkan definisi, algoritma serta *pseudocode* algoritma *bubble sort.* Kemudian kelas *TutorialActivity* akan memandu pengguna menjalani tutorial cara mengurutkan angka acak yang telah tersedia.

Kelas *GameActivity* dieksekusi ketika pengguna memilih mulai bermain pada kelas *MainActivity.* Pada kelas *GameActivity* akan ditampilkan beberapa angka yang masih teracak dan pengguna diminta untuk menukar angka acak dengan cara men-*drag and drop* angka yang tersedia menurut metode *bubble sort*. Jika langkah yang dilakukan pengguna benar maka posisi angka akan tertukar. Jika langkah yang dilakukan pengguna salah maka posisi angka tidak tertukar dan akan ditampilkan teks bantuan untuk membantu pengguna mengurutkan angka yang benar.

Kelas *BubbleSort* merupakan kelas yang berisi atribut dan operasi untuk menginisialisasi, membandingkan dan mengurutkan angka yang sedang

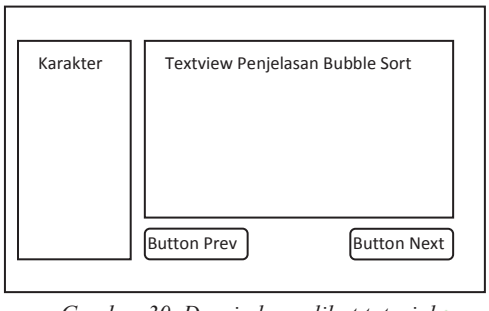


Gambar 29. Desain layar halaman utama

diurut oleh pengguna. Setiap langkah pengurutan oleh pengguna, maka objek kelas *BubbleSort* akan membandingkan angka yang dipilih oleh pengguna.

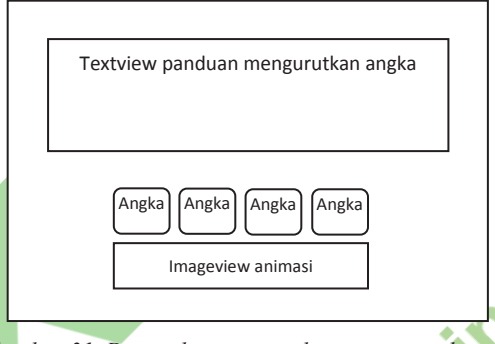
- ***Perancangan User Interface***

Pada layar halaman utama terdapat titel aplikasi *bubble sort* dan 2 tombol yang dapat dipilih oleh pengguna yaitu tombol mulai bermain yang akan memanggil kelas *GameActivity* jika pengguna ingin mulai bermain mengurutkan angka dan tombol lihat tutorial yang akan memanggil kelas *TutorialActivity*.



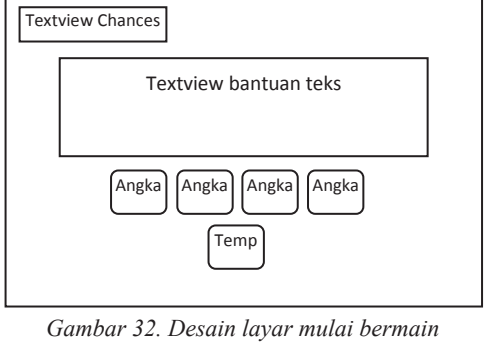
Gambar 30. Desain layar lihat tutorial

Pada saat pengguna menjalankan tutorial, kelas *TutorialActivity* akan menampilkan teks penjelasan definisi bubble sort dan *pseudocode* algoritma bubble sort. Tombol *prev* dan *next* berfungsi sebagai navigasi untuk melihat teks penjelasan sebelumnya atau teks penjelasan berikutnya.



Gambar 31. Desain layar tutorial mengurut angka

Layar tutorial mengurut angka akan ditampilkan setelah layar teks penjelasan algoritma *bubble sort* pada gambar 31. Pengguna akan dipandu oleh teks dan animasi untuk mengurutkan angka acak yang tersedia.



Gambar 32. Desain layar mulai bermain

Pada layar bermain, pengguna dapat mulai mengurutkan angka acak yang tersedia, tiap langkah pengurutan yang dilakukan oleh pengguna, maka aplikasi akan memberikan teks hasil jika langkah pengurutan benar atau salah.

3. Pengujian

- ***Spesifikasi Perangkat Pengujian***

Spesifikasi perangkat yang disarankan untuk dapat menjalankan aplikasi permainan untuk pembelajaran *bubble sort* sebagai berikut: (1) *Smartphone* dengan sistem operasi *Android; (2).* Layar 4.5 *inches; dan (3)* Sistem operasi *Android* versi 4.0 *(Ice Cream Sandwich)* keatas

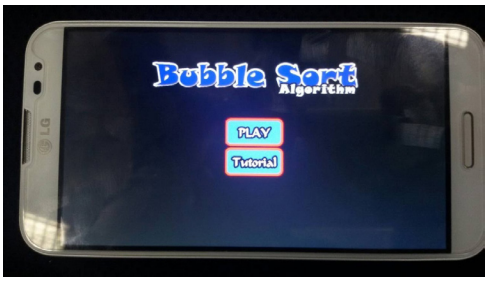
- ***Hasil Pengujian Aplikasi***

a. Tampilan Layar Utama

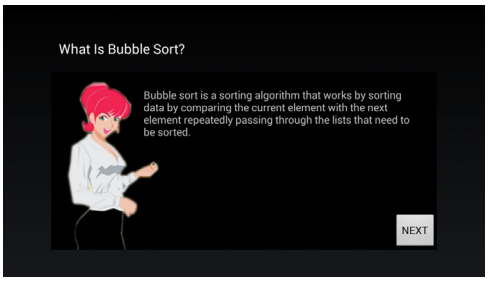
Pada saat aplikasi dijalankan, tampilan layar utama pada gambar 34 akan ditampilkan. Pada layar utama terdapat dua menu yang dapat dipilih oleh pengguna yaitu menu *play* dan menu tutorial. Jika pengguna menjalankan menu *play* maka aplikasi akan menampilkan layar bermain dan pengguna dapat melakukan aktivitas mulai bermain. Jika pengguna menjalankan menu tutorial maka aplikasi akan menampilkan layar tutorial yang dapat dilihat oleh pengguna.

Gambar 33. Tampilan layar utama

Pengujian fungsi-fungsi aplikasi dilakukan pada dua perangkat yang berbeda seperti pada gambar 35 dan gambar 36.



Gambar 34. Tampilan layar utama pada perangkat 5.5 Inch



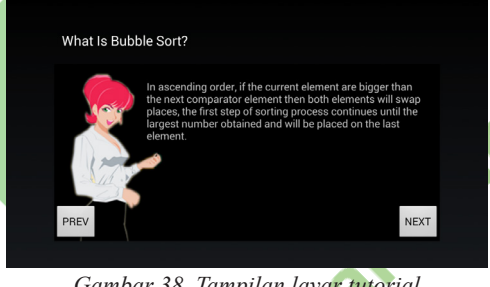
Gambar 35. Tampilan layar utama pada perangkat 7 Inch

b. Tampilan Layar Tutorial

Layar tutorial ditampilkan setelah pengguna memilih tombol tutorial pada layar utama. Pada saat tutorial dijalankan, pengguna dapat membaca teks penjelasan mengenai definisi dan *pseudocode* algoritma *bubble sort* serta dipandu cara bermain mengurutkan angka. pengguna dapat menekan tombol *next* untuk melihat teks penjelasan selanjutnya.

Gambar 36. Tampilan layar tutorial

Setelah tombol *next* ditekan, pengguna dapat membaca teks penjelasan lanjutan definisi *bubble sort.* Pengguna juga dapat kembali ke tampilan penjelasan sebelumnya dengan menekan tombol *prev* seperti pada gambar 37.

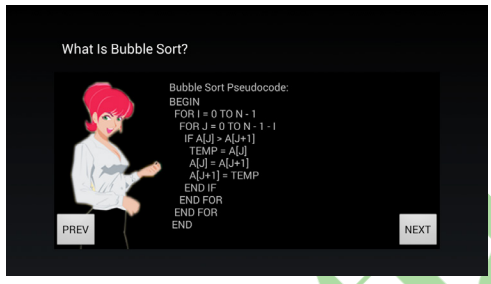


Gambar 38. Tampilan layar tutorial

Layar selanjutnya masih akan menampilkan teks penjelasan lanjutan definisi *bubble sort* ketika pengguna menekan tombol *next* pada layar seperti pada gambar 38.

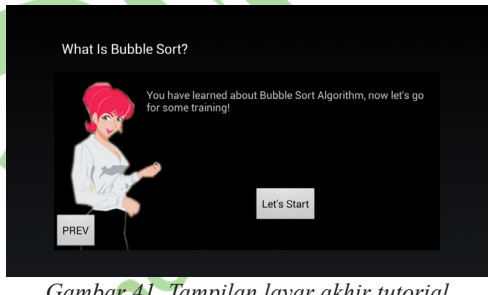
Gambar 39. Tampilan layar tutorial

Tampilan *pseudocode* algoritma *bubble sort* pada gambar 40 ditampilkan setelah tampilan teks penjelasan definisi *bubble sort* dan pengguna menekan tombol *next* pada layar tutorial.



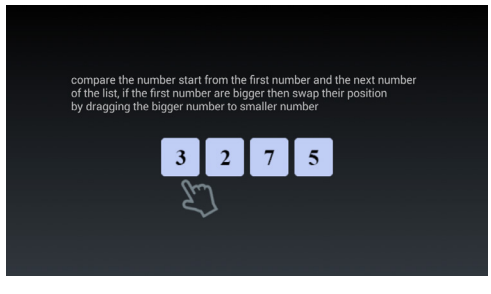
Gambar 40. Tampilan pseudocode pada layar tutorial

Setelah semua teks penjelasan definisi dan *pseudocode* algoritma *bubble sort* ditampilkan, pengguna akan diarahkan kepada layar panduan bermain mengurutkan angka dengan metode *bubble sort.* Pengguna dapat menekan tombol *Let's Start* seperti pada gambar 41.



Gambar 41. Tampilan layar akhir tutorial

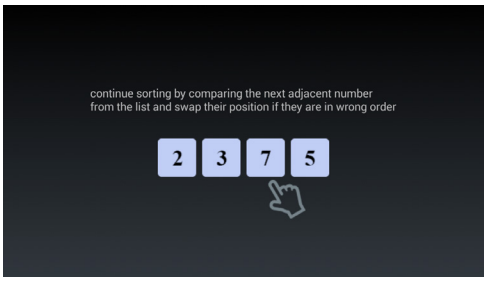
Pada layar panduan bermain, pengguna dapat membaca teks petunjuk cara bermain dan melihat objek animasi berupa tangan yang bergerak memandu pengguna untuk men-*drag and drop* angka sesuai dengan yang ditunjuk oleh objek animasi.



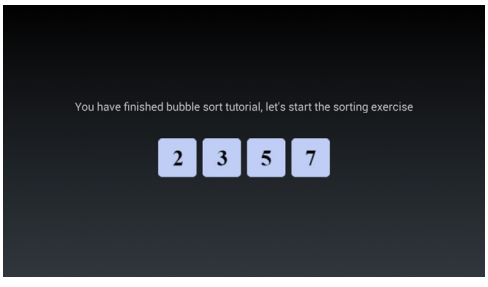
Gambar 42. Tampilan layar panduan bermain

Setelah pengguna men-*drag and drop* sesuai dengan panduan teks dan animasi maka posisi kedua angka akan tertukar seperti pada gambar 43. Selanjutnya objek animasi tangan akan memandu pengguna untuk men-*drag and drop* angka berikutnya.

Setelah semua angka terurut, aplikasi akan menampilkan notifikasi bahwa tutorial telah selesai dan pengguna dapat mulai bermain mengurutkan angka. Tampilan layar dapat dilihat pada gambar 44.



Gambar 43. Tampilan layar panduan bermain



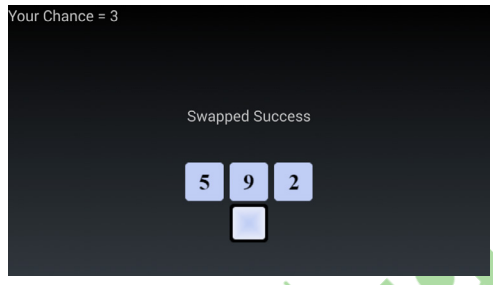
Gambar 44. Tampilan layar panduan bermain

c.Tampilan Layar Bermain

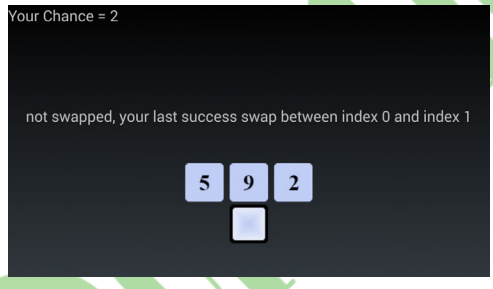
Layar bermain ditampilkan setelah pengguna menekan tombol *play* pada tampilan layar utama aplikasi pada gambar 34. Pada saat mulai bermain, aplikasi akan menampilkan 3 angka acak. Pengguna hanya memiliki 3 kesempatan dalam melakukan kesalahan langkah pengurutan. Jika pengguna melakukan 3 kali kesalahan mengurut angka pada saat bermain, maka permainan akan di-*reset* mulai dari awal. Status kesempatan pengguna ditampilkan pada pojok kiri atas layar seperti pada gambar 45. Kotak kosong yang berada di bawah angka berfungsi untuk menampilkan angka yang disimpan pada variabel *temp* dalam algoritma *bubble sort*. Ketika pengguna menekan dan men-*drag* angka maka kotak *temp* akan menampilkan angka yang sedang ditekan.

Gambar 45. Tampilan layar bermain

Jika langkah pengurutan yang dilakukan pengguna benar maka aplikasi akan menampilkan pesan bahwa angka telah berhasil ditukar seperti pada gambar 46. Jika pengguna melakukan kesalahan langkah dalam mengurutkan angka, maka kesempatan pengguna akan dikurangi satu dan teks bantuan akan ditampilkan untuk membantu pengguna mengingat angka terakhir yang bertukar posisi sehingga pengguna dapat mencari angka selanjutnya yang akan ditukar.



Gambar 46. Tampilan pengurutan berhasil pada layar bermain

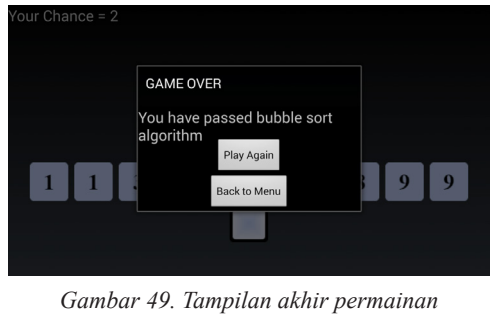


Gambar 47. Tampilan pengurutan salah pada layar bermain

Jika semua angka telah terurut, aplikasi akan menaikkan level dengan kembali menampilkan angka acak dengan jumlah angka yang bertambah satu angka menjadi 4 angka acak seperti pada gambar 48. Jika kesempatan pengguna telah habis karena kesalahan langkah pengurutan maka permainan akan di-*reset* kembali pada level awal dengan jumlah 3 angka acak.

Gambar 48. Tampilan layar bermain pada level berikutnya

Pengguna dapat melanjutkan permainan mengurut angka hingga level sepuluh angka acak. Jika pengguna berhasil mengurut angka hingga level sepuluh angka acak maka permainan akan berakhir



Gambar 49. Tampilan akhir permainan

dan pengguna memiliki pilihan untuk mengulang permainan dari level awal atau kembali ke tampilan layar utama aplikasi seperti pada gambar 49.

4. Taksonomi Dalam Permainan

Berdasarkan model pembelajaran taksonomi Bloom yang digunakan dalam penelitian ini untuk mengukur proses mempelajari algoritma maka aplikasi permainan untuk pembelajaran algoritma *bubble sort* dapat mengakomodasi beberapa tingkatan dalam model pembelajaran taksonomi Bloom sebagai berikut: (1) Aplikasi permainan memiliki teks penjelasan algoritma *bubble sort* yang dapat dibaca oleh pengguna sebelum mulai bermain; (2) Pengguna dapat memperhatikan *flowchart* algoritma *bubble sort* dan tutorial *game* agar dapat memahami proses algoritma *bubble sort*; (3) Dalam proses bermain, pengguna dapat menerapkan pengetahuan yang telah dibaca pada tutorial aplikasi untuk menyelesaikan *game*; dan (4) Pengguna dapat menganalisis setiap langkah algoritma *bubble sort* untuk memenangkan *game*.

IV. SIMPULAN

Dari hasil dan evaluasi pada bab 4 maka dapat disimpulkan beberapa hal:

- Aplikasi permainan pengurutan *bubble sort* memiliki penjelasan singkat mengenai algoritma *bubble sort* untuk mengakomodasi tingkat *knowledge* dalam taksonomi Bloom.
- Aplikasi pengurutan *bubble sort* memiliki tutorial cara bermain.
- Aplikasi permainan pengurutan *bubble sort* dapat digunakan sebagai alat bantu untuk melatih kemampuan individu dalam mempelajari algoritma *bubble sort*.

VI. DAFTAR PUSTAKA

[1] M. Zarlis & Handrizal, "Algoritma dan Pemrograman: Teori dan Praktik dalam Pascal," Medan, USU Press, 2008, p. 1.

[2] R. A. S & M. Shalahuddin, Modul Pembelajaran Algoritma dan Pemrograman, Bandung: Modula, 2010.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithm 3rd Edition, London: The MIT Press, 2009.

[4] D. Suryani, "Perbandingan Metode Bubble Sort dan Insertion Sort Terhadap Efisiensi Memori," Jurnal Teknologi Informasi dan Pendidikan, vol. 6, p. 2, 2013.

[5] R. Rheinadi, "Analisis Algoritma Bubble Sort," Makalah IF2091 Strategi Algoritmik, 2009.

[6] F. Yuwono, "Algoritma Postman's Sort Untuk 4 Digit Dengan Bahasa Pemrograman Pascal," 2003.

[7] R. Lawlor, "Dublin Institute of Technology," [Online]. Available: http://www.comp.dit.ie/rlawlor/Alg_DS/sorting/Bubble%20Sort.pdf. [Accessed 25 mei 2014].

[8] "Analisis Algoritma Bubble Sort," 11 juni 2012. [Online]. Available: http://www.mesran.net/berita-493-analisis-algoritma-bubble-sort.html. [Accessed 25 mei 2014].

[9] T. Hakim, "Belajar dan Prinsip Belajar," in Belajar Secara Efektif, Jakarta, Puspa Swara, 2000, p. 1.

[10] A. Suprijono, Cooperative Learning: Teori dan Aplikasi PAIKEM, Surabaya: Pustaka Pelajar, 2009.

[11] S. Shabanah & D. J. X. Chen, "Simplifying Algorithm Learning Using Serious Games".

[12] S. E. Utami, "Algoritma," in 10 Langkah Belajar Logika dan Algoritma Menggunakan Bahasa C dan C++ di GNU/Linux, Yogyakarta, ANDI, 2005, p. 19.

[13] A. H. Sutopo, "M-Learning," in Teknologi Informasi dan Komunikasi dalam Pendidikan, Yogyakarta, Graha Ilmu, 2012, p. 175.

[14] A. Ismail, Education Games, Yogyakarta: Pro U Media, 2009, p. 27.

[15] M. Zyda, "From Visual Simulation To Virtual Reality To Games," p. 25, 2005.

[16] J. Smed & H. Hakonen, "Towards a Definition of a Computer Game," September 2003.

[17] "Cambridge Dictionaries Online," Cambridge University Press, 2015. [Online]. Available: http://dictionary.cambridge.org/dictionary/british/computer-game. [Accessed 12 Agustus 2015].

[18] Y. Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle," International Journal of Engineering and Technology, vol. 2, 2012.

[19] S. Dharwiyanti & R. S. Wahono, Pengantar Unified Modeling Language (UML), IlmuKomputer.com, 2003.