

## **PENGEMBANGAN PERMAINAN “RUN BOBBY! FOOD!” DENGAN MENGGUNAKAN MICROSOFT KINECT**

### ***DEVELOPING "RUN BOBBY! FOOD!" USING MICROSOFT KINECT***

Yohan Muliono<sup>1</sup>, Handy<sup>2</sup>, Ellys<sup>3</sup>, Kelvin Chandra<sup>4</sup>, Rhio Sutoyo<sup>5</sup>

<sup>1</sup>Cyber Security Program, Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

<sup>2,3,4,5</sup>Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

<sup>1</sup>ymuliono@binus.edu, <sup>2</sup>handytedja@gmail.com, <sup>3</sup>huang.ellys@gmail.com,  
<sup>4</sup>kelvinchandra1024@gmail.com, <sup>5</sup>rsutoyo@binus.edu

#### **Abstrak**

Tujuan dari penelitian ini adalah untuk menghasilkan sebuah permainan pada *platform* PC dengan menggunakan teknologi Microsoft Kinect. Teknologi ini digunakan untuk menyediakan pengalaman bermain yang lebih baik sehingga bisa lebih menarik perhatian pengunjung pameran. Microsoft Kinect digunakan untuk mendeteksi gerakan tubuh manusia sebagai pemain. Selain itu, Kinect juga digunakan sebagai media interaksi antara aplikasi dengan pengguna. Metode penelitian yang digunakan dibagi menjadi identifikasi masalah, pengumpulan data, perancangan aplikasi, implementasi, dan evaluasi. Pengumpulan data dilakukan dengan kuesioner, observasi, studi pustaka, dan wawancara kepada ahli. Perancangan aplikasi “Run Bobby! Food!” dilakukan dengan metode perancangan aplikasi scrum. Evaluasi dilakukan dengan dua pendekatan, yaitu kuesioner dan wawancara kepada ahli. Hasil dari penelitian ini adalah 96,1% responden puas dengan aplikasi.

**Kata Kunci:** *computer vision, exhibition, kinect, object detection, scrum*

#### **Abstract**

*The purpose of this research is to produce a game on a PC platform using Microsoft Kinect technology. This technology is used to provide a better playing experience to attract more visitors to the exhibition. Microsoft Kinect was utilized to detect the movement of the human body (i.e. players) used as a medium of interaction between the applications and users. Research method in this research was divided into problem identification, data collection, application design, implementation, and evaluation. The data were collected through questionnaires, observations, literature studies, and interview with experts. Application design “Run Bobby! Run!” was done by scrum method. The evaluation was done by applying two approaches, namely questionnaires and interview with experts. The result of this research shows 96.1% responders were satisfied with the application. With these promising results, it could be concluded that the application “Run Bobby! Run!” was suitable for exhibition at Bina Nusantara University to attract more visitors.*

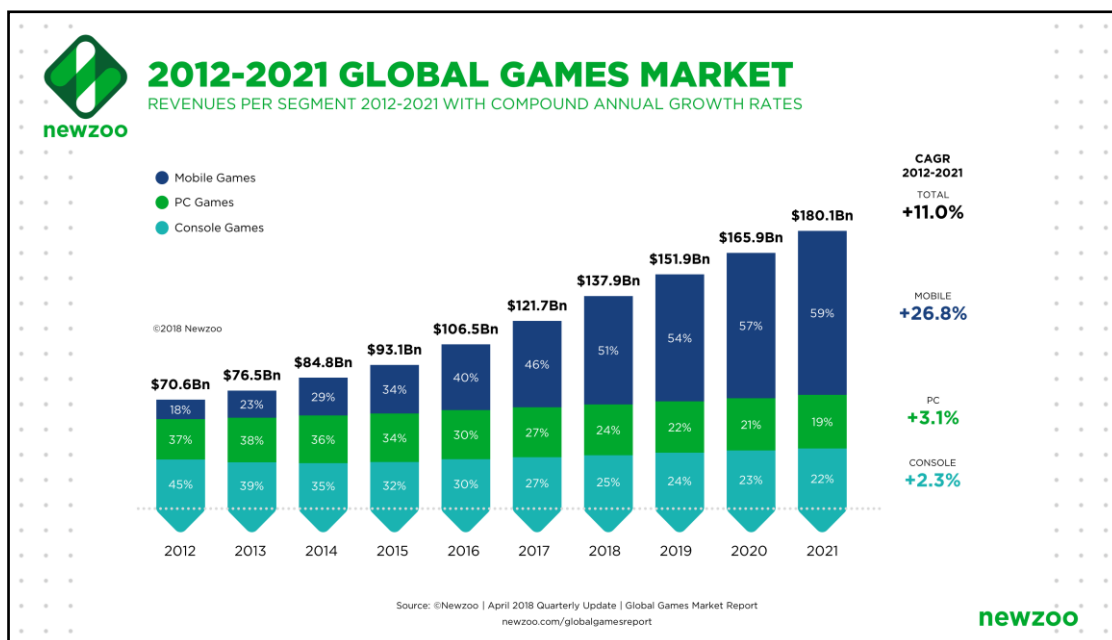
*Keywords: computer vision, exhibition, Kinect, object detection, scrum*

**Tanggal Terima Naskah : 06 Juli 2018**

**Tanggal Persetujuan Naskah : 16 Juli 2018**

## 1. PENDAHULUAN

Dalam pengembangan *game* yang menarik, Microsoft Kinect merupakan salah satu solusi yang tepat [1]. Hal ini dikarenakan *input* yang diterima oleh alat tersebut melibatkan hampir semua anggota tubuh manusia. Hal ini diharapkan dapat lebih menarik perhatian pengunjung. Kinect merupakan salah satu alat deteksi gerakan dan objek dari Microsoft yang dapat dipakai untuk membuat suatu aplikasi [2], [3] dan permainan [4], [5]. Tipe program yang dirancang dan dibuat adalah permainan. Pemilihan ini dikarenakan permainan adalah bisnis yang sangat sukses dan juga populer di mata masyarakat dunia. Permainan sudah ada selama bertahun-tahun dan menghasilkan milyaran dolar [6]. Pada tahun 2012, pendapatan industri permainan dari seluruh dunia mencapai USD 70,6 milyar. Jumlah tersebut telah meningkat hampir sebesar 100% pada tahun 2018. Pendapatan bisnis dari *video game* diperkirakan akan mencapai hingga USD 180,1 milyar di tahun 2021. Rincian perkembangan pendapatan *video game* tersebut dapat dilihat pada Gambar 1 berikut ini.



Gambar 1. Statistik Pendapatan Industri *Video Game* [6]

Permainan yang dibuat bertema *endless running*. Hal ini dikarenakan tipe permainan ini melibatkan seluruh anggota tubuh, yaitu dengan cara berlari untuk menggerakkan avatar di dalam permainan. Selain itu, permainan dengan tipe ini masih populer di kalangan masyarakat. Hal ini terbukti dengan adanya permainan Subway Surfers dari Kiloo yang ada dalam peringkat 10 besar permainan gratis di Google Play [7]. Permainan ini juga dapat memberikan perhitungan kalori yang dibakar atau dikeluarkan oleh pemain setelah menyelesaikan suatu tahap.

Pada penelitian sebelumnya permainan digunakan sebagai alat yang mengajarkan pemain untuk selalu berbuat baik dengan memberikan hadiah jika pemain memilih untuk berbuat baik di dalam permainan [8]. Pada aplikasi lain, *Fix Our Environment*, permainan yang mengajarkan pemain untuk merawat lingkungan untuk menghindari bencana [9]. Selain itu, terdapat juga permainan yang mengajarkan mengenai materi pembelajaran mengenai ilmu pengetahuan alam [10]. Melihat penelitian sebelumnya yang berhasil dilakukan, sudah terbukti bahwa permainan dapat mengajarkan pemain untuk melakukan hal positif yang telah mereka pelajari di dunia permainan ke dalam dunia nyata.

Dengan merancang permainan ini, hasil yang ditargetkan adalah menghasilkan permainan yang dapat menarik minat pengunjung pameran dan dapat membuat pemainnya melakukan gerakan yang mirip dengan olahraga lari. Dengan fitur-fitur pada permainan, seperti *high score* dan *instant calories calculation feedback*, diharapkan permainan ini dapat membuat membuat olahraga berlari menjadi menarik untuk dilakukan.

## 2. KONSEP DASAR

### 2.1 *Game Balancing*

*Game balancing* adalah konsep untuk membuat permainan menjadi tidak terlalu mudah ataupun terlalu sulit, dan harus dirasa adil oleh para pemain ataupun individu pemain itu sendiri, supaya permainan terasa lebih menyenangkan [11]. Berdasarkan lawan pemain, permainan dibagi menjadi dua, yaitu *player-versus-player* (PvP) dimana lawan pemain adalah pemain lain, bahkan jika pemain tersebut bersifat artifisial, dan *player-versus-environment* (PvE). Untuk membuat sebuah permainan yang seimbang, terdapat beberapa teknik yang dapat digunakan, antara lain hindari strategi dominan, menggunakan elemen peluang, membuat permainan PvP adil, membuat permainan PvE adil, mengatur tingkat kesulitan, dan mengerti umpan balik positif. Selain teknik-teknik tersebut, tingkat kesulitan dinamis pun bisa menjadi pilihan untuk menciptakan permainan yang menarik dan seimbang. Permainan dengan tingkat kesulitan yang dinamis akan menyesuaikan tingkat kesulitan permainan dengan keterampilan pemain sehingga permainan tidak akan terlalu mudah atau sulit bagi pemain. Keunggulan dari mode ini adalah pemain selalu mendapatkan pengalaman baru setiap kali bermain [12].

### 2.2 Microsoft Kinect

Menurut Tsai, Kuo, Chu, dan Yen, sensor Kinect merupakan sensor gerakan yang menyediakan *natural user interface* [13]. Kinect ditemukan dan dikembangkan oleh Microsoft yang pada awalnya digunakan untuk konsol Xbox 360 [1]. Kinect dapat memberikan pengalaman baru untuk mengontrol Xbox 360 karena menggunakan gerakan tubuh dan pengenalan suara. Kinect menggunakan kamera untuk dapat mendeteksi kerangka/skeleton dari manusia agar dapat digunakan untuk *motion detection*. Saat ini, Kinect v2 for Windows/Kinect Xbox One juga memiliki kemampuan lebih untuk mendeteksi *depth* atau kedalaman, merekam *video high definition* (HD) hingga 1080p, jangkauan area pendeteksian yang lebih luas, dan kemampuan *infrared* yang lebih tinggi.

### 2.3 Genre Permainan

*Action Game* menggunakan keterampilan fisik pemain sebagai tantangan sedangkan *Arcade Game* adalah *Action Game* yang didesain dengan model bisnis untuk mendapatkan uang dimana pemain harus memasukkan koin ke mesin permainan [11]. *Arcade Game* secara umum tidak bisa dimenangkan dan semakin lama semakin susah hingga pemain kalah. Hal ini memotivasi pemain untuk memasukkan lebih banyak koin.

### 2.4 Pembakaran Kalori

Terdapat tiga faktor penentu dalam perhitungan jumlah kalori terbakar. *Metabolic Equivalent* (MET) merupakan variabel yang menandakan tingkat metabolisme dari mengkonsumsi 3,5 mililiter oksigen per kilogram per menit [14], [15].

$$\text{Kalori Terbakar} = \text{MET} \times \text{Berat (kg)} \times \text{Waktu (jam)} \dots\dots\dots (1)$$

MET memiliki nilai yang berbeda-beda untuk setiap aktivitas yang dilakukan. Menurut daftar nilai MET pada American Time Use Survei (ATUS) yang dipublikasikan di website Applied Research Cancer Control and Population Sciences, aktivitas berlari memiliki nilai MET 7,5. Sedangkan untuk berat, diambil berat rata-rata seluruh manusia di dunia, yaitu 62 kg. Waktu dihitung bukan dari lamanya pemain memainkan permainan, melainkan dari berapa jumlah langkah yang ditempuh pemain dibagi dengan jumlah langkah rata-rata manusia dalam dalam satu jam.

### 3. METODE PERANCANGAN

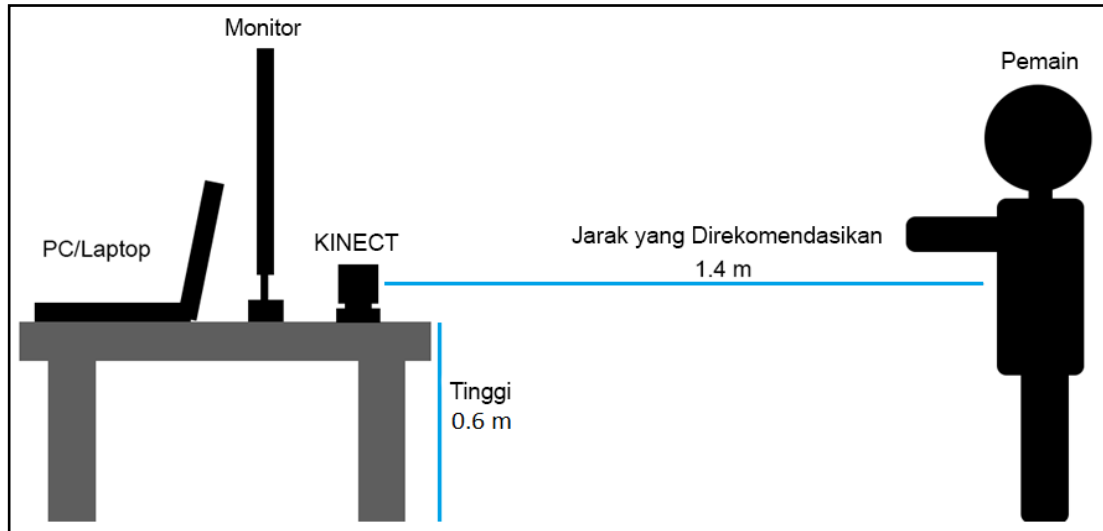
Metode perancangan yang dipakai dalam mengembangkan permainan ini diawali dengan pengumpulan data, yaitu berupa kuesioner, melakukan *literature review* terhadap beberapa penelitian terdahulu dan teknologi yang dibutuhkan, serta melakukan wawancara kepada ahli. Kuesioner terdiri atas 10 pertanyaan yang bertujuan untuk mencari tahu minat pengunjung terhadap aplikasi atau permainan yang ditampilkan pada pameran. Hasil dari kuesioner ini mempengaruhi hasil akhir dari permainan yang dikembangkan. Wawancara dilakukan kepada salah satu dosen Universitas Bina Nusantara yang sering membantu pameran sebagai ahli.

Dari hasil kuesioner dan wawancara yang dilakukan, permainan yang dibuat akan menggunakan Microsoft Kinect sebagai alat pendeteksi *input*. Hal ini dikarenakan Microsoft Kinect, yang menggunakan hampir seluruh anggota badan sebagai alat *input*, diharapkan dapat menarik pengunjung untuk bermain ke area pameran. Aplikasi dibungkus dengan permainan berlari yang dapat menghitung kalori yang dibakar. Pemilihan tema olahraga ini diharapkan dapat menjadi daya tarik tersendiri bagi permainan ini. Setelah didapatkan kebutuhan pengguna, perancangan permainan dibuat dan disertai dengan tampilan gambar dan perancangan sistem basis data.

#### 3.1 Perancangan Permainan

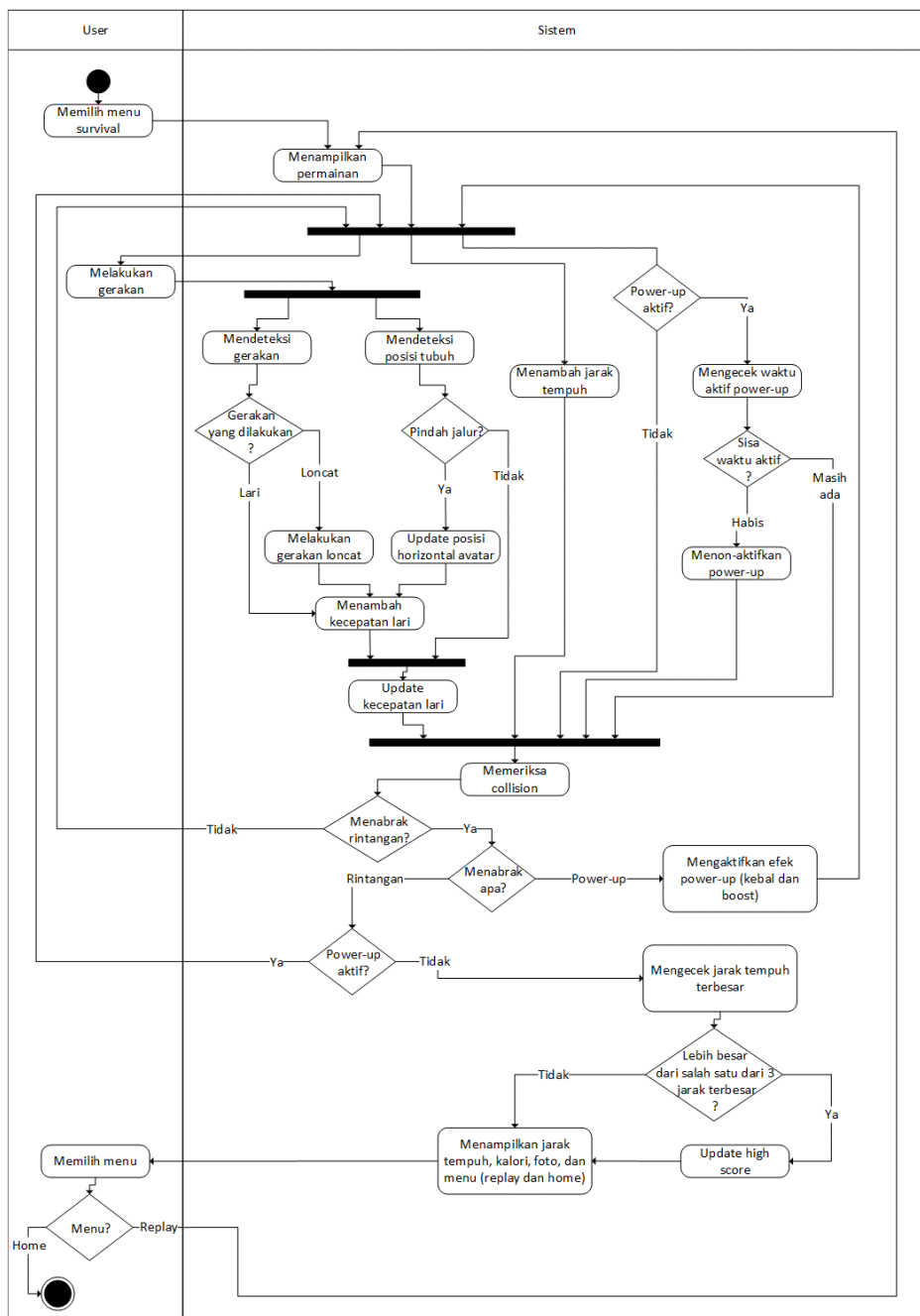
Run Bobby! Food! merupakan permainan *endless run* dengan *genre action* yang dimainkan menggunakan Microsoft Kinect. Permainan *endless run* ini bertema makanan. Karakter pemain berlari di atas jalan bertekstur pizza yang dibatasi pagar donat di bagian kiri dan kanannya. Di sepanjang tepi jalan, terdapat es krim, lollipop, dan sosis. Terdapat banyak rintangan yang harus dilewati pemain di tengah jalan. Selain rintangan, juga terdapat *power-up* yang berbentuk *energy drink* dimana jika diambil maka karakter akan mendapatkan efek tembus rintangan dan lari cepat selama lima detik.

Permainan ini dirancang untuk *platform PC/Laptop. Monitor* dan Microsoft Kinect dibutuhkan untuk menampilkan permainan dan menangkap *input* dari pemain. Untuk mendapat pengalaman bermain yang maksimal, area bermain harus memiliki penerangan yang cukup terang dan tersebar rata pada satu ruangan. Selain itu, jarak dari pemain ke Kinect minimal 1,4 meter untuk satu pemain. Dalam permainan ini harus dipastikan juga tidak boleh ada benda yang menghalangi sensor Kinect. Perangkat Kinect disarankan untuk diletakkan di atas layar. Pembuatan aplikasi ini menggunakan Unity Video Game Engine. Unity adalah sebuah aplikasi pengembangan permainan yang gratis dan cukup populer digunakan di dunia pengembang permainan [16], [17].



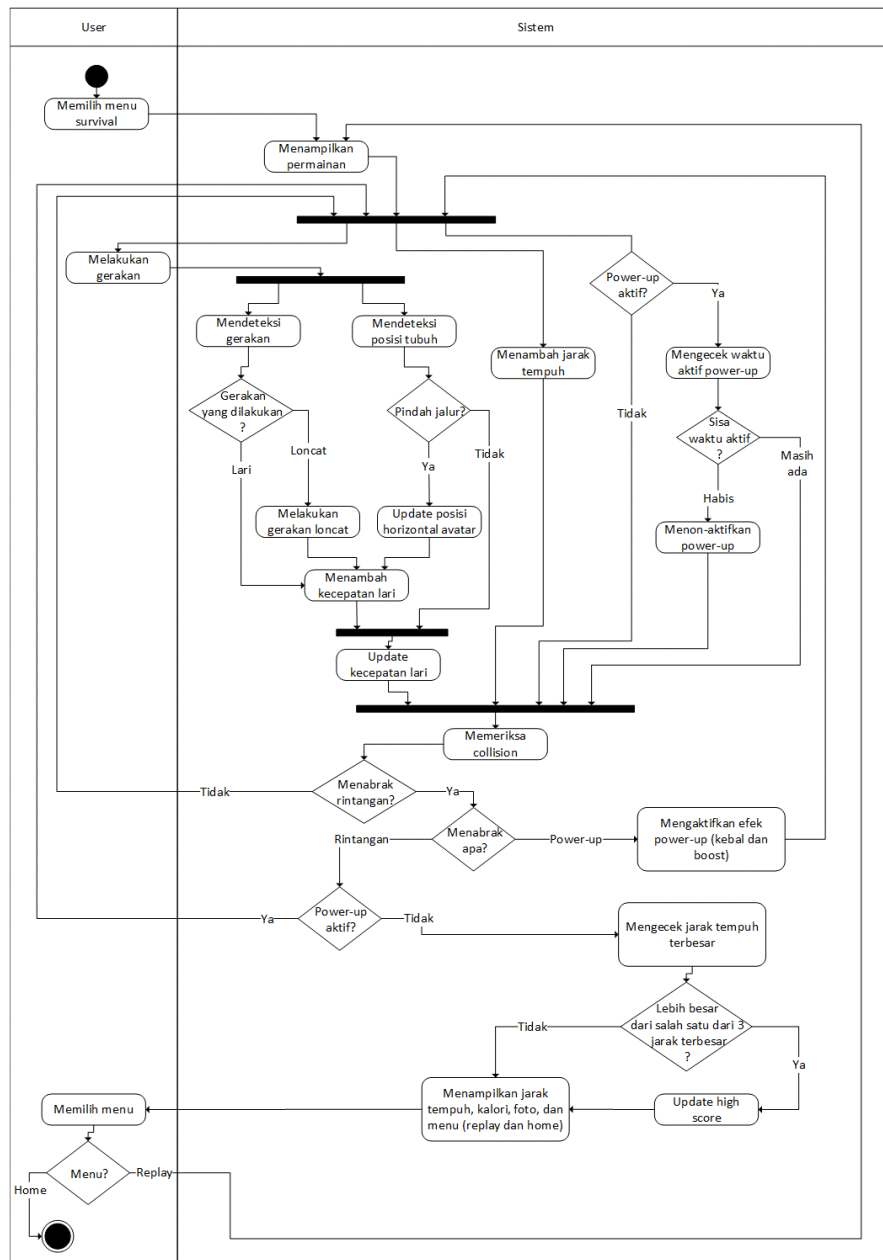
Gambar 2. Rancangan peletakan Peralatan dan Pemain

Permainan terbagi menjadi dua *mode*, *mode survival* dan *mode time attack*. *Mode survival* dapat dimainkan tanpa batas. Skor akan dihitung berdasarkan jarak yang ditempuh dalam permainan. Permainan akan berhenti apabila rata-rata kecepatan lari pemain dalam 5 detik di bawah batas minimal atau karakter menabrak rintangan. Ketika memilih menu *Survival*, pemain akan langsung diarahkan ke dalam permainan. Tanda “RUN!” akan dimunculkan selama satu detik dan setelah itu, pemain akan mendapatkan kontrol untuk bermain. Pemain harus berlari untuk mengisi bar yang ada di sebelah kanan bawah layar. Dengan bergeser ke arah kiri atau kanan, seperti yang ditunjukkan oleh bar berwarna biru muda di sebelah bar kecepatan, karakter dapat berpindah jalur untuk menghindari rintangan yang ada.



Gambar 3. Activity Diagram permainan Mode Survival

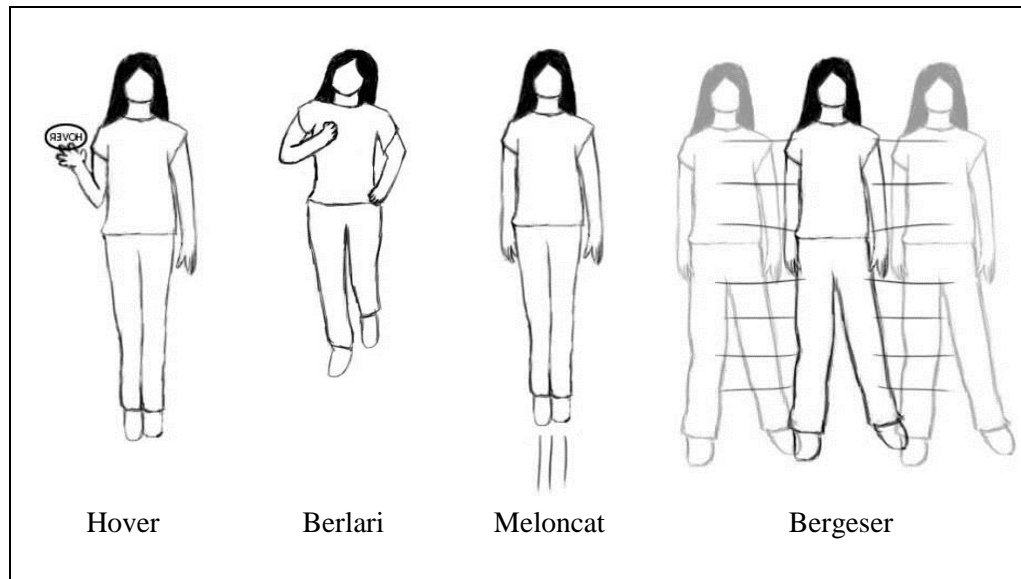
Pada *mode time attack*, pemain harus berlari sejauh jarak yang telah dipilih. Terdapat tiga pilihan jarak untuk *mode* ini, yaitu 200 meter, 400 meter, dan 600 meter dimana satuan meter merupakan satuan dalam permainan, bukan satuan sebagaimana aslinya. Skor yang akan dihitung berupa waktu yang dibutuhkan untuk menempuh jarak yang telah dipilih. Kecepatan lari karakter bergantung pada rata-rata kecepatan lari pemain dalam 5 detik. Permainan akan berhenti apabila karakter menabrak rintangan. Terdapat tiga jalur pada *track* lari pemain dan banyak rintangan yang harus dilewati, juga *monster* yang mengejar karakter.



Gambar 4. Activity Diagram permainan Mode Time Attack

Pemain dapat melihat *high score* yang didapatkan selama bermain. *High score* yang akan ditampilkan adalah tiga skor terbesar untuk *mode survival* dan tiga waktu tercepat untuk masing-masing jarak di *mode time attack*. Pada permainan Run Bobby! Food!, diperlukan gerakan tubuh sebagai interaksi untuk bermain dan navigasi pada menu. Gerakan yang digunakan pada permainan ada empat macam, yaitu:

- a. *Hover* : Menahan kursor di satu tempat pada waktu tertentu
- b. Berlari : Melangkah dengan kecepatan yang tinggi
- c. Meloncat : Melompat dengan menggunakan kedua kaki secara bersamaan
- d. Bergeser : Menggeser badan ke kiri atau ke kanan dari posisi awal



Gambar 5. Jenis-jenis aksi yang dapat dilakukan pada aplikasi

Karakter akan terus berlari selama permainan berjalan. Rata-rata kecepatan lari juga dapat ditambahkan dengan cara meloncat atau berpindah jalur. Terdapat rintangan-rintangan yang harus dilewati oleh pemain dengan cara meloncat atau berpindah jalur. Selain itu, terdapat *item power-up* yang akan membuat karakter menjadi *invisible* (kebal terhadap rintangan) dan berlari cepat selama 5 detik. Selama *mode power-up* ini, karakter masih bisa meloncat atau berpindah jalur. Ketika permainan berakhir, pemain dapat memilih untuk bermain lagi atau pindah ke menu awal.

### 3.2 Perancangan Basis Data

Data pada permainan, seperti *high score* dan *setting* disimpan di dalam *file* yang diberi nama “*savedGames.gd*”. *File* tidak dapat dibuka secara langsung, melainkan dengan cara melakukan *deserialize* dengan struktur *file* yang sesuai pada saat melakukan *serialize* ke dalam *file*.

Tabel 1. Struktur Data Dalam *savedGames.gd*

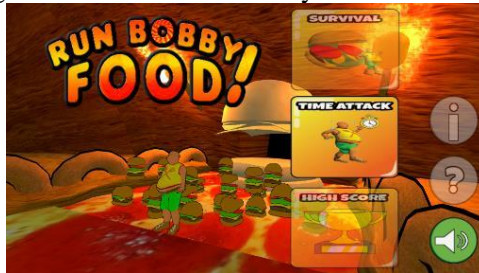
Attribute	Description	Data Type
<i>HighScoreSurvival</i>	Menyimpan tiga jarak terjauh dari <i>game mode survival</i>	List<float>
<i>HighScoreTimeAttack</i>	Menyimpan tiga waktu terkecil dalam menyelesaikan <i>game mode time attack</i>	List<float>
<i>isSoundEnabled</i>	Menyimpan konfigurasi untuk suara aktif dalam <i>game</i>	bool



## 4. HASIL DAN PEMBAHASAN

### 4.1 Hasil Implementasi

Run Bobby! Food! terbagi menjadi empat bagian, yaitu menu utama, permainan *mode survival*, permainan *mode time attack*, *high score*, dan *tutorial*. Untuk dapat pindah ke menu yang diinginkan, pemain harus melakukan *hover* terhadap gambar menu yang diinginkan tersebut. Pemain juga dapat melihat *credits* dengan memilih gambar menu dengan huruf I atau menyalakan atau mematikan suara pada permainan dengan memilih gambar dengan lambang suara dimatikan atau dinyalakan.



Gambar 6. Tampilan Menu Utama Run Bobby! Food!

*Mode survival* dimainkan dengan tujuan menempuh jarak terjauh yang ada pada *high score*. Ketika memilih menu *Survival*, pemain akan langsung diarahkan ke dalam permainan. Tanda “RUN!” akan dimunculkan selama satu detik dan setelah itu, pemain akan mendapatkan kontrol untuk bermain. Pemain harus berlari untuk mengisi *bar* yang ada di sebelah kanan bawah layar. Dengan bergeser ke arah kiri atau kanan, seperti yang ditunjukkan oleh *bar* berwarna biru muda di sebelah *bar* kecepatan, karakter dapat berpindah jalur untuk menghindari rintangan yang ada. Selain itu, pemain juga dapat meloncat untuk menghindari rintangan yang ada di depan karakter. Jarak yang telah ditempuh oleh karakter ditampilkan pada tengah atas layar permainan. Ketika karakter mendapatkan *power-up*, karakter akan diselimuti oleh cahaya dan lingkaran yang menunjukkan waktu aktif *power-up* akan muncul di bawah tampilan jarak. Permainan akan berakhir apabila karakter menabrak rintangan atau tinggi kecepatan rata-rata berada di bawah batas minimal.



Gambar 7. Tampilan Permainan *Mode Survival*



Gambar 8. Tampilan Setelah Permainan *Mode Survival* Berakhir

Ketika permainan berakhir, Kinect akan mengambil sebuah foto untuk ditampilkan pada layar berakhirnya permainan. Foto tersebut diletakkan di bagian kiri layar dan tidak akan disimpan ke dalam sistem dalam bentuk apapun. Jarak yang ditempuh dan kalori yang terbakar oleh pemain akan ditampilkan di sebelah kanan layar bersamaan dengan gambar yang dapat dilakukan *hover* oleh pemain untuk mengulang permainan atau kembali ke

menu utama. Apabila jarak yang ditempuh oleh pemain saat itu mengalahkan salah satu dari tiga jarak terjauh yang tercatat pada *high score*, maka akan muncul notifikasi “New High Score” di atas foto yang ditampilkan dan jarak tersebut akan tersimpan pada *high score*.

*Mode time attack* dimainkan dengan tujuan menempuh jarak tertentu dengan waktu sesingkat-singkatnya. Ketika memilih menu *time attack*, pemain akan diarahkan untuk memilih jarak yang akan dimainkannya. Pemain juga dapat melihat waktu tercepat untuk menyelesaikan jarak yang akan dipilihnya. Jika pemain tidak jadi bermain dengan *mode time attack*, pemain dapat melakukan *hover* terhadap *button back* yang ada pada kiri atas layar pemilihan jarak.



Gambar 9. Tampilan memilih jarak untuk *Mode Time Attack*



Gambar 10. Tampilan setelah permainan *Mode Time Attack* berakhir

Setelah memilih jarak, pemain akan diarahkan ke dalam permainan. Tanda “RUN!” akan dimunculkan selama satu detik dan setelah itu, pemain akan mendapatkan kontrol untuk bermain. Pemain harus berlari untuk mengisi *bar* yang ada di sebelah kanan bawah layar. Pada *mode* ini, semakin cepat pemain berlari, maka kecepatan karakter untuk menempuh jarak yang diperlukan akan meningkat. Dengan bergeser ke arah kiri atau kanan, seperti yang ditunjukkan oleh *bar* berwarna biru muda di sebelah *bar* kecepatan, karakter dapat berpindah jalur untuk menghindari rintangan yang ada. Selain itu, pemain juga dapat melompat untuk menghindari rintangan yang ada di depan karakter.

Tampilan sisa jarak akan ditampilkan di tengah atas layar permainan dan tampilan waktu yang telah digunakan akan diletakkan di kanan atas layar permainan. Ketika karakter mendapatkan *power-up*, karakter akan diselimuti oleh cahaya dan lingkaran yang menunjukkan waktu aktif *power-up* akan muncul di bawah tampilan sisa jarak. Permainan akan berakhir apabila karakter menabrak rintangan atau jarak yang dipilihnya sudah habis ditempuh.

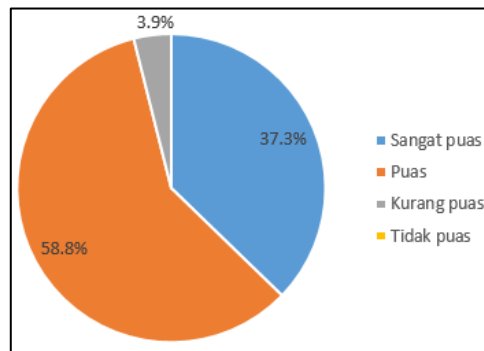
## 4.2 Evaluasi

Tahap evaluasi dilakukan dengan membuka demo aplikasi kepada mahasiswa dan kemudian memberikan kuesioner yang berisikan 13 pertanyaan untuk diisi setelah pemain mencoba aplikasi. Demo aplikasi dilakukan pada tanggal 4 Januari 2016 dan 9 Januari 2016 di Universitas Bina Nusantara ruang 708. Pada akhir demo aplikasi, didapatkan 51 responden yang telah mencoba aplikasi dan mengisi kuesioner.

Berikut adalah *list* dari pertanyaan yang ditanyakan kepada *user* yang diminta untuk mengevaluasi permainan Run Bobby! Food! Memberikan pengalaman baru dalam bermain game *endless run*?

- Apakah *game* ini menarik perhatian anda?
- Apakah *game* ini mudah dimainkan (*Mode Survival*)?
- Apakah *game* ini mudah dimainkan (*Mode Time Attack*)?
- Apakah tampilan pada *game* ini mudah dimengerti?
- Apakah tampilan pada *game* ini konsisten dan tidak membingungkan?

- f. Apakah menu *tutorial* membantu anda untuk memainkan *game* ini?
- g. Apakah gerakan yang digunakan untuk berinteraksi dengan *game* ini mudah diingat dan dilakukan?
- h. Seberapa cepat Anda dapat menguasai kontrol pada *game* ini?
- i. Apakah *game* ini memberikan umpan balik yang sesuai dan informatif?
- j. Seberapa sulit tantangan yang ditawarkan di dalam *game* ini?
- k. Bagaimana kinerja *game* ini ketika dimainkan?
- l. Apakah Anda puas dengan *game* ini?



Gambar 11. Grafik jawaban mengenai kepuasan pengguna terhadap aplikasi

Berdasarkan hasil kuesioner yang telah disebar, sebanyak 19 orang menjawab bahwa mereka sangat puas dengan permainan Run Bobby! Food!, 30 orang menjawab bahwa mereka puas dengan permainan ini, dua orang menjawab bahwa mereka kurang puas dengan permainan ini, dan tidak ada yang menjawab bahwa mereka tidak puas dengan permainan ini. Dengan demikian, dapat disimpulkan bahwa pemain puas dengan permainan Run Bobby! Food!

## 5. KESIMPULAN

Berdasarkan analisis, perancangan, implementasi, dan evaluasi yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi Run Bobby! Food! dapat menarik perhatian pengunjung pameran hasil karya mahasiswa Universitas Bina Nusantara dengan gerakan-gerakan yang digunakan untuk menggunakan aplikasi. Gerakan-gerakan tersebut, antara lain *hover* untuk menggerakkan *cursor*, *hold* untuk memilih menu yang tersedia, lari untuk membuat karakter berlari, loncat untuk membuat karakter melompat, dan bergeser ke kanan atau ke kiri untuk membuat karakter berpindah jalur. Hasil kepuasan pengguna terhadap aplikasi adalah 96,1% pengguna merasa puas dengan kualitas aplikasi yang dihasilkan.

## REFERENSI

- [1] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE Multimed.*, vol. 19, no. 2, pp. 4–10, 2012.
- [2] A. Cordar, A. Robb, A. Wendling, S. Lampotang, C. White, and B. Lok, “Virtual Role-Models: Using Virtual Humans to Train Best Communication Practices for Healthcare Teams,” vol. 9238, no. September, 2015, pp. 229–238.
- [3] R. Sutoyo *et al.*, “KINECTATION (Kinect for Presentation): Control Presentation with Interactive Board and Record Presentation with Live Capture Tools,” in *Journal of Physics: Conference Series*, 2017, vol. 801, no. 1, p. 12053.
- [4] E. M. Suyanto, D. Angkasa, H. Turaga, and R. Sutoyo, “Overcome Acrophobia with the Help of Virtual Reality and Kinect Technology,” *Procedia Comput. Sci.*, vol.

- 116, pp. 476–483, 2017.
- [5] M. Zhang, Z. Zhang, Y. Chang, E.-S. Aziz, S. Esche, and C. Chassapis, “Recent Developments in Game-Based Virtual Reality Educational Laboratories Using the Microsoft Kinect,” *Int. J. Emerg. Technol. Learn.*, vol. 13, no. 1, pp. 138–159, 2018.
- [6] T. Wijman, “Mobile Revenues Account for More Than 50% of the Global Games Market as It Reaches \$137.9 Billion in 2018,” *Newzoo*, 2018. [Online]. Available: <https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/>. [Accessed: 28-Jun-2018].
- [7] “Subway Surfers.” [Online]. Available: <https://play.google.com/store/apps/details?id=com.kiloo.subwaysurf>. [Accessed: 28-Jun-2018].
- [8] D. David, D. Diana, F. A. Luwinda, and D. S. Widasro, “Game" The Genuine" dengan Dynamic Branching Story,” *Tek. dan Ilmu Komput.*, vol. 6, no. 24, 2017.
- [9] F. R. Tijan, A. T. Rianto, N. Afina, V. Yesmaya, and F. A. Luwinda, “Game" Fix Our Environment" Applications Design Based on Android,” *Tek. dan Ilmu Komput.*, vol. 5, no. 19, 2016.
- [10] H. Ham, R. Budhiarjo, H. Giantoko, and A. Agung, “PERANGKAT AJAR ILMU PENGETAHUAN ALAM TINGKAT SEKOLAH DASAR ‘BUMI SAHABATKU’ PADA SISTEM OPERASI IOS MENGGUNAKAN COCOS2DX,” *Tek. dan Ilmu Komput.*, vol. 7, no. 26, 2018.
- [11] E. Adams, *Fundamentals of Game Design*, 3rd ed. Thousand Oaks, CA, USA: New Riders Publishing, 2014.
- [12] R. Sutoyo, D. Winata, K. Oliviani, and D. M. D. M. Supriyadi, “Dynamic Difficulty Adjustment in Tower Defence,” in *Procedia Computer Science*, 2015, vol. 59, pp. 435–444.
- [13] C.-H. Tsai, Y.-H. Kuo, K.-C. Chu, and J.-C. Yen, “Development and evaluation of game-based learning system using the Microsoft Kinect sensor,” *Int. J. Distrib. Sens. Networks*, vol. 11, no. 7, p. 498560, 2015.
- [14] N. M. Byrne, A. P. Hills, G. R. Hunter, R. L. Weinsier, and Y. Schutz, “Metabolic equivalent: one size does not fit all,” *J. Appl. Physiol.*, vol. 99, no. 3, pp. 1112–1119, 2005.
- [15] S. Kozey, K. Lyden, J. Staudenmayer, and P. Freedson, “Errors in MET estimates of physical activities using  $3.5 \text{ ml} \cdot \text{kg}^{-1} \cdot \text{min}^{-1}$  as the baseline oxygen consumption,” *J. Phys. Act. Heal.*, vol. 7, no. 4, pp. 508–516, 2010.
- [16] W. Goldstone, *Unity Game Development Essentials*. Packt Publishing, 2009.
- [17] S. Blackman, *Beginning 3D Game Development with Unity: All-in-one, Multi-platform Game Development*, 1st ed. Berkely, CA, USA: Apress, 2011.