

Pelatihan Advanced Angry Bird Game Programming Untuk Siswa SD Pada Pusat Kegiatan Belajar Masyarakat (PKBM) Semarang

Indra Gamayanto¹, Sendi Novianto², Arta Moro Sundjaja³

^{1,2}Fakultas Ilmu Komputer, Departemen Sistem Informasi & Teknik Informatika, Universitas Dian Nuswantoro (UDINUS), Iman Bonjol 207, Semarang

³Departemen Manajemen, BINUS Business School, Master Program, Bina Nusantara University (BINUS), Jakarta

E-mail: ¹indra.gamayanto@dsn.dinus.ac.id, ²sendi.novianto@dsn.dinus.ac.id, ³asundjaja@binus.edu

Abstrak

Game merupakan sesuatu yang menarik, dimana dengan game banyak orang yang termotivasi untuk belajar dan hal ini akan dapat meningkatkan kreativitas. Masalah seperti belum tersedianya infrastruktur untuk menunjang pembuatan game dan sumber daya manusia yang masih terbatas akan dapat diatasi dengan pelatihan khusus. Pada pengabdian masyarakat ini, kami akan berfokus untuk memberikan pelatihan kepada siswa agar dapat membuat game yang sederhana dan hal ini dapat dikembangkan menjadi sebuah game yang menarik di masa depan. Masih banyak sekali, peluang yang akan didapat oleh siswa jika dapat menerapkan game ini karena ini akan dapat membantu siswa dalam meningkatkan daya ingat, kreativitas dan inovasinya. Hasil dari pengabdian masyarakat ini adalah siswa akan dapat membuat sebuah game yang sederhana melalui bahasa programming yang telah diajarkan, dan kemudian akan mengimplementasikannya ke dalam beberapa pelajaran di PKBM, sehingga hal ini akan dapat bermanfaat bagi masa depan siswa.

Kata kunci: Game, Programming, Siswa, Implementasi, Inovasi

Abstract

Games are something interesting, where with games many people are motivated to learn and this will increase creativity. Problems such as the unavailability of infrastructure to support game creation and limited human resources can be overcome with special training. In this community service, we will focus on providing training to students so they can create simple games and this can be developed into an interesting game in the future. There are still many opportunities that students will get if they can apply this game because it will help students improve their memory, creativity and innovation. The result of this community service is that students will be able to create a simple game using the programming language that has been taught, and will then implement it into several lessons at PKBM, so that this will be useful for the students' future.

Keywords: Game, Programming, Student, Implementation, Innovation

1. PENDAHULUAN

Belajar programming dengan membuat game merupakan sesuatu yang sangat menarik dan dapat terus ditingkatkan sehingga anak-anak akan dapat merasakan pengalaman yang berbeda dalam belajar meningkatkan kemampuannya. Metode pembelajaran ini sangat berguna agar game dapat menjadi salah satu sarana pembelajaran terutama agar mempermudah dalam memahami programming. Dalam pengabdian ini, kami semua berfokus dalam mengembangkan game ke

tingkat yang lebih jauh, dimana pembelajaran ini akan dapat menghasilkan anak-anak yang dapat menggunakan bahasa pemrograman tertentu tetapi tentunya dengan cara yang lebih baik.

Terdapat beberapa alasan penting kenapa game sangat baik dalam pengembangan kompetensi. Game membutuhkan keterampilan utama antara lain: pembuatan sprite dan efek suara diperlukan, dimana pembuatan game membutuhkan pembelajaran cara menulis kode dan cara menggunakan engine atau framework yang digunakan. Yang paling memakan waktu dari ketiganya adalah membuat aset baru. Sampai game tersebut selesai dan sesuai dengan desainnya sendiri, biasanya ada cukup banyak grafik dan model gratis yang tersedia untuk meminimalkan hal ini[1]. Programmer dapat dengan mudah memahami engine dan kode. Pengembang perangkat lunak yang memasuki desain game harus mempelajari teknologi dan memikirkan konten imajinatif yang akan membuat game menjadi menyenangkan. Pembuatan game lebih dari sekadar pengodean[2]. Karena semakin banyak perusahaan mencari programmer berpengalaman dengan keterampilan C++, DirectX, HTML5, dan dasar 2D mobile, pengembangan game memiliki keunggulan di luar sektor game[3]. Pengembangan gim video biasanya lebih mudah jika dilakukan secara bertahap. Desainer gim harus memulai dari yang kecil. Setelah beberapa minggu atau bulan, Unity atau Unreal memungkinkan pembuatan gim dalam hitungan hari atau. Penguji gim lebih mudah ditemukan daripada penguji aplikasi proyek sampingan, oleh karena itu dapat memperoleh umpan balik yang lebih informatif. Menguji gim pada berbagai fase pengembangan sama prosesnya dengan mengerjakan aplikasi lainnya[4]. Unreal dan Unity memungkinkan programmer berkonsentrasi pada desain game sementara mesin menangani pencahayaan, perilaku pemain, dan rendering grafis. DirectX, OpenGL, dan JavaScript/WebGL berjalan pada Unreal Engine 4 dan Unity 5[5]. Unreal dan Unity memudahkan pengembangan kode untuk game 3D atau 2D pada berbagai konsol, PC/Mac, dan perangkat seluler, apa pun bahasa atau platformnya. Alasan keempat kami mempelajari pemrograman game adalah karena mesin game, yang menggerakkan game dan memberdayakan programmer, tidak pernah lebih kuat, lebih cepat, lebih tangkas, lebih cekatan, atau lebih mudah digunakan. Kita dapat menjual dan menghasilkan uang dari game yang dibuat seperti halnya menjual aplikasi[6]. Monetisasi sering kali gratis, tetapi hambatan untuk masuk mungkin lebih besar daripada aplikasi karena pasar game cenderung lebih teregulasi. Setiap platform yang mendistribusikan game pasti sudah memiliki fitur seperti mekanisme pembayaran dan pustaka yang sudah dibangun, tidak seperti aplikasi, yang terkadang mengharuskan membuat pustaka sendiri[7]. Cukup dengan meluncurkan game dan menyebarkannya, dan hal ini mudah dilakukan. Kenyataan menunjukkan bahwa para gamer memberikan sebuah kombinasi permainan yang dapat meningkatkan kenaikan level dengan cara membuat perencanaan terlebih dahulu untuk dapat memenangkan game tersebut[8]. Anak-anak dapat memahami dan mempelajari pemrograman gim dan menulis beberapa kode ke dalam gim yang akan memungkinkan untuk dilakukan. Hal terakhir untuk belajar menulis perangkat lunak pengembangan gim adalah kesempatan untuk mendorong generasi gamer berikutnya untuk mengikuti jejak yang sudah dilakukan dan menjadi programmer[9]. Berlawanan dengan persepsi umum, membuat game dapat membantu mempelajari rekayasa perangkat lunak. Terutama jika belajar dengan tujuan. Jika telah bekerja dengan bahasa berorientasi objek sepanjang karier. Di sini mungkin menginginkan mesin dengan sistem komponen yang komprehensif. Aktivitas ini akan mengajarkan desain berbasis komponen, yang dapat membantu meningkatkan sistem kerja. Jika mencari mesin dan kerangka kerja yang berbeda di sini akan diperoleh banyak hal yang bermanfaat. Ada hal lain yang dipertaruhkan di sini selain kesenangan dan permainan. Jika ingin memajukan karier, mempelajari pengembangan game adalah cara yang tepat untuk melakukannya[10]. Anggap saja sedang mengerjakan proyek web satu halaman berbasis React saat ini, tetapi sama sekali tidak mengenal kerangka kerja tersebut dan tidak tahu cara memulai mempelajarinya sendiri. Salah satu pilihannya adalah mendesain game dan membangun situs web di React (atau kerangka kerja pilihan) untuk menyimpan dan menyinkronkan data game seperti skor tertinggi dan kemajuan. Server yang dikomunikasikan oleh klien game mungkin dibangun

menggunakan teknologi yang sama jika tim hanya bekerja di sisi server[11]. Hal terpenting yang perlu diingat adalah bahwa desain permainan dapat membantu kemajuan karier dan pendidikan. Lebih jauh lagi, budaya pengembangan game adalah bagian favorit. Kelompok dan pertemuan pengembang game sangat bagus untuk membangun jaringan dan kolaborasi proyek[12]. Pembuatan Game itu menyenangkan, sama seperti halnya pengembangan perangkat lunak lainnya, di sini mungkin mengalami frustrasi, kurangnya arahan, dan mungkin perasaan berjuang melawan alat yang ada. Pembuatan gim video bisa jadi menyenangkan. Memainkan dan menyempurnakan gim merupakan bagian penting dari proses pengembangan, dan tidak ada yang lebih baik daripada menyelesaikan dan merilis gim[13].

Permasalahan seperti sulitnya melatih bahasa pemrograman kepada anak-anak akan terjawab di dalam pengabdian masyarakat ini. Hasil dari pengabdian masyarakat ini adalah anak-anak akan dapat memiliki kompetensi dalam mengembangkan kemampuan programmingnya, sehingga akan dapat meningkatkan kualitas generasi berikutnya di masa depan.

2. METODE



Gambar 1. Proses pengabdian masyarakat

Gambar 1 menjelaskan tahapan dalam pengabdian masyarakat. Tahap pertama, kita memperkenalkan apa itu bahasa pemrograman. Tahap kedua adalah kita membuat game dan terakhir adalah sebuah proyek yang menghasilkan game.

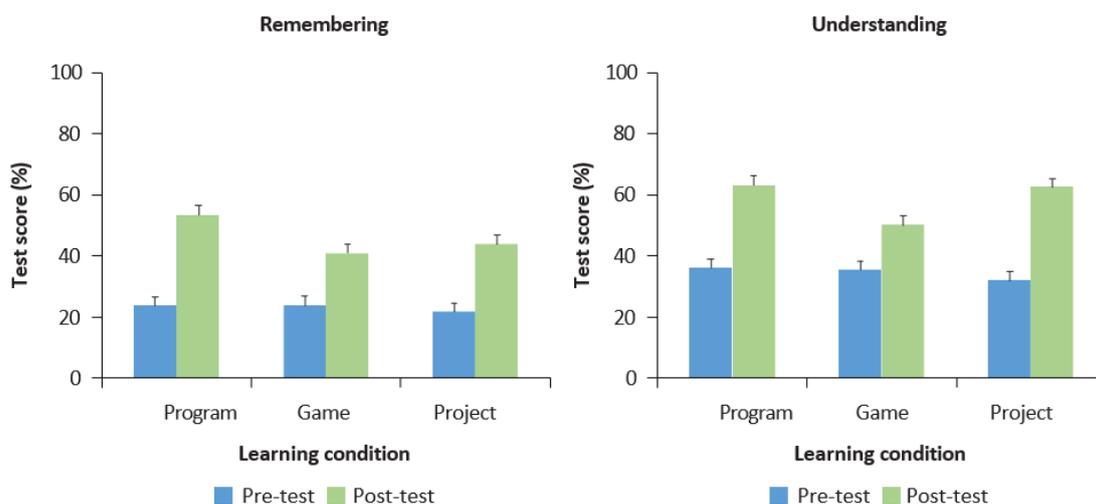
3. HASIL DAN PEMBAHASAN

3.1. Kegiatan pengabdian masyarakat



Gambar 2. Kegiatan pengabdian masyarakat

3.2. Pre-Test & Post Test



Gambar 3. Pre-Test & Post Test

Gambar 3 menjelaskan pre test, dimana anak-anak pada awalnya masih memiliki kebingungan dalam memahami bahasa pemrograman, tetapi setelah pelatihan dimulai, anak-anak mulai memahami bahasa pemrograman. Kami kemudian melanjutkannya ke tahap berikutnya yaitu melatih anak-anak dalam membuat game, hasilnya pada post test anak-anak sudah memahami

dan mampu dalam membuat game

3.3. Pemrograman dan game

3.3.1. Angry bird game

```
import pygame
import math

# Initialize pygame
pygame.init()

# Constants
WIDTH, HEIGHT = 800, 600
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)

screen = pygame.display.set_mode((WIDTH, HEIGHT))
clock = pygame.time.Clock()

def distance(p1, p2):
    return math.sqrt((p1[0] - p2[0]) ** 2 + (p1[1] - p2[1]) ** 2)

# Bird class
class Bird:
    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.radius = 15
        self.color = RED
        self.dragging = False
        self.velocity = [0, 0]
        self.launched = False

    def draw(self, screen):
        pygame.draw.circle(screen, self.color, (self.x, self.y), self.radius)

    def update(self):
        if self.launched:
            self.x += self.velocity[0]
            self.y += self.velocity[1]
            self.velocity[1] += 0.5 # Gravity

# Main loop
bird = Bird(150, 450)
running = True
while running:
    screen.fill(WHITE)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
```

```
running = False
elif event.type == pygame.MOUSEBUTTONDOWN:
    if distance((bird.x, bird.y), pygame.mouse.get_pos()) < bird.radius:
        bird.dragging = True
elif event.type == pygame.MOUSEBUTTONUP:
    if bird.dragging:
        bird.dragging = False
        bird.launched = True
        mouse_x, mouse_y = pygame.mouse.get_pos()
        bird.velocity = [(bird.x - mouse_x) / 5, (bird.y - mouse_y) / 5]

if bird.dragging:
    bird.x, bird.y = pygame.mouse.get_pos()

bird.update()
bird.draw(screen)
pygame.display.flip()
clock.tick(60)
```

```
pygame.quit()
```

3.3.2. Game dan masa depan

Pemrograman untuk desainer game

Berikut adalah lima keuntungan penting mempelajari cara membuat kode:

1. Pembuatan portofolio. Membuat portofolio sebagai desainer game pemula bisa jadi menantang, tetapi dengan keterampilan pemrograman dasar, Anda dapat mengembangkan game kecil untuk memamerkan keterampilan dan ambisi Anda. Meskipun game tersebut mungkin bukan game berkualitas AAA, game tersebut tetap merupakan tambahan yang berharga untuk portofolio Anda[13],[14].
2. Peningkatan keterampilan Desain Game. Menurut pengalaman saya. Setelah beberapa bulan mempelajari pemrograman, kualitas dokumentasi desain saya meningkat secara signifikan. Pengodean juga membantu saya berpikir tentang desain sistem tingkat rendah dan bagaimana mekanisme dapat bekerja sama, sehingga menghasilkan peningkatan kualitas kerja secara keseluruhan[15],[16].
3. Pengembangan pemikiran sistem. Pengodean melibatkan pemecahan masalah yang rumit menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola. Proses ini dapat membantu mengembangkan keterampilan berpikir sistem Anda, yang berguna saat Anda akan mengerjakan sistem dan mekanisme permainan yang rumit dan rumit[17],[18].
4. Peningkatan komunikasi. Programmer adalah orang yang paling sering berkomunikasi dengan desainer game. Mengetahui cara membuat kode dapat membantu Anda menyampaikan ide kepada para insinyur dengan lebih baik, memahami cara kerja mereka, dan cara mengatasi beberapa keterbatasan teknis dengan cara yang cerdas[19].
5. Meningkatkan pembuatan prototipe. Pengetahuan pemrograman sangat penting untuk pembuatan prototipe cepat dalam pengembangan game. Pengetahuan ini mempercepat proses dan meningkatkan kualitas prototipe. Prototipe bukti konsep yang kecil juga dapat dengan cepat menunjukkan hasil yang diinginkan dan melengkapi dokumentasi desain game[20].

6. Pengembangan game indie. Pengembang game indie memerlukan keterampilan pemrograman karena mereka bekerja dalam tim kecil, terkadang bahkan sendirian. Pemrograman sangat penting karena setiap anggota tim memiliki banyak tanggung jawab. Bahkan dengan seorang programmer, mungkin tidak ada cukup waktu untuk menerapkan atau membuat prototipe beberapa mekanisme[20].

Berikut ini adalah tabel untuk menjelaskan jenis-jenis pengembang game, antara lain:

Tabel 1. Jenis-jenis pengembang game

Jenis	Peran	Contoh
Pengembang Game	Mengembangkan game secara internal untuk satu platform atau konsol	Naughty Dog (Sony), Retro Studios (Nintendo), Obsidian Entertainment (Xbox Game Studios)
Pengembang pihak kedua	Mengembangkan game secara eksklusif untuk konsol (tetapi tidak sepenuhnya dimiliki oleh platform atau konsol tersebut)	Game Freak (mengembangkan Pokémon untuk Nintendo tetapi tidak sepenuhnya dimiliki oleh Nintendo)
Pengembang pihak ketiga	Mengembangkan dan menerbitkan game mereka sendiri atau mengembangkan game untuk penerbit game lain	Ubisoft, Seni Elektronik, Activision Blizzard
Pengembang indie	Mengembangkan sebagian besar game indie yang didanai sendiri sebagai individu atau kelompok kecil	Eric Barone, Thelka

4. KESIMPULAN DAN SARAN

4.1. KESIMPULAN

Kesimpulan yang dapat diambil setelah melakukan pengabdian masyarakat adalah sebagai berikut:

1. Game merupakan sarana yang baik dalam melatih anak-anak dalam belajar programming. Hal ini karena bahasa pemrograman yang tadinya dirasa rumit akan terasa mudah pada saat menggunakan sarana yang menyenangkan
2. Pelatihan ini akan dapat meningkatkan kompetensi anak-anak dalam memahami bahasa pemrograman sebagai bahasa yang dapat digunakan untuk banyak hal yang bermanfaat. Oleh karena itu pembelajaran seperti ini harus sering dilakukan untuk dapat meningkatkan kompetensi
3. Meningkatkan pembelajaran menggunakan bahasa pemrograman membutuhkan konsistensi dan pelatihan yang tepat sehingga hal ini dapat dicapai dengan efektif dan efisien

4.2. SARAN

Saran yang dapat diberikan setelah melakukan pengabdian masyarakat, antara lain:

1. Pelatihan model seperti ini dapat diterapkan dalam banyak kurikulum dan subjek mata pelajaran. Oleh karena itu, model seperti ini dapat dilakukan secara berkala sehingga mempersiapkan anak-anak dimulai dari kecil untuk dapat belajar bahasa programming

2. Pembelajaran tipe ini membutuhkan waktu yang cukup lama tetapi hasilnya akan dapat meningkatkan kompetensi secara baik dan secara berangsur-angsur akan dapat memberikan banyak manfaat
3. Perlu disediakan infrastruktur teknologi informasi di setiap sekolah agar dapat menerapkan model pembelajaran yang demikian secara konsisten

UCAPAN TERIMA KASIH

Terima kasih kepada semua pihak yang telah membantu terselenggaranya pengabdian masyarakat ini. Kami sangat berterima kasih kepada (PKBM) Semarang, yang telah memberikan kesempatan kepada kami untuk dapat melakukan pengabdian masyarakat sehingga dapat membantu meningkatkan kompetensi generasi berikutnya demi masa depan Indonesia

DAFTAR PUSTAKA

- [1] “A Guideline for Game Development-Based Learning: A Literature Review - Wu - 2012 - International Journal of Computer Games Technology - Wiley Online Library.” Accessed: Mar. 07, 2025. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1155/2012/103710>
- [2] E. Bethke, *Game Development and Production*. Wordware Publishing, Inc., 2003.
- [3] “Game development life cycle guidelines | IEEE Conference Publication | IEEE Xplore.” Accessed: Mar. 07, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6761558>
- [4] S. Rabin, *Introduction To Game Development (Game Development)*. USA: Charles River Media, Inc., 2005.
- [5] J. Novak, *Game Development Essentials: An Introduction (4th Edition)*. NOVY PUBLISHING, 2022.
- [6] C. M. Kanode and H. M. Haddad, “Software Engineering Challenges in Game Development,” in *2009 Sixth International Conference on Information Technology: New Generations*, Apr. 2009, pp. 260–265. doi: 10.1109/ITNG.2009.74.
- [7] J. Habgood and M. Overmars, *The Game Maker’s Apprentice: Game Development for Beginners*. Apress, 2006.
- [8] B. Berg Marklund, H. Engström, M. Hellkvist, and P. Backlund, “What Empirically Based Research Tells Us About Game Development,” *Comput Game J*, vol. 8, no. 3, pp. 179–198, Dec. 2019, doi: 10.1007/s40869-019-00085-1.
- [9] M. Zhu and A. I. Wang, “Model-driven Game Development: A Literature Review,” *ACM Comput. Surv.*, vol. 52, no. 6, p. 123:1-123:32, Nov. 2019, doi: 10.1145/3365000.
- [10] S. Aslan and O. Balci, “GAMED: digital educational game development methodology,” *SIMULATION*, vol. 91, no. 4, pp. 307–319, Apr. 2015, doi: 10.1177/0037549715572673.
- [11] W. Goldstone, *Unity Game Development Essentials*. Packt Publishing Ltd, 2009.
- [12] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, “What went wrong? A survey of problems in game development,” *Comput. Entertain.*, vol. 7, no. 1, p. 13:1-13:22, Feb. 2009, doi: 10.1145/1486508.1486521.
- [13] B. Kenwright, “Holistic game development curriculum,” in *SIGGRAPH ASIA 2016 Symposium on Education*, in SA ’16. New York, NY, USA: Association for Computing Machinery, Nov. 2016, pp. 1–5. doi: 10.1145/2993352.2993354.
- [14] R. Owston, H. Wideman, N. S. Ronda, and C. Brown, “Computer game development as a literacy activity,” *Computers & Education*, vol. 53, no. 3, pp. 977–989, Nov. 2009, doi: 10.1016/j.compedu.2009.05.015.

- [15] B. Bergeron, *Developing Serious Games (Game Development Series)*. USA: Charles River Media, Inc., 2005.
- [16] S. Kurkovsky, "Engaging students through mobile game development," *SIGCSE Bull.*, vol. 41, no. 1, pp. 44–48, Mar. 2009, doi: 10.1145/1539024.1508881.
- [17] A. Chia, B. Keogh, D. Leorke, and B. Nicoll, "Platformisation in game development," *Internet Policy Review*, vol. 9, no. 4, Art. no. 4, Oct. 2020.
- [18] E. Westecott, "Independent game development as craft," *Loading... The Journal of the Canadian Game Studies Association*, vol. 7, no. 11, Art. no. 11, 2013.
- [19] C. Keith, *Agile Game Development with Scrum*. Pearson Education, 2010.
- [20] J. Hocking, *Unity in Action, Third Edition: Multiplatform Game Development in C#*. Simon and Schuster, 2022.