

PEMBUATAN GAME ARCADE “CITY PROTECTOR” DENGAN MENGGUNAKAN TOUCH GESTURE RECOGNIZER

Andrean Danawan¹⁾ Jeanny Pragantha²⁾ Darius Andana Haris³⁾

^{1) 2) 3)} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta
email: andreandanawan@gmail.com¹⁾, jeanny@fti.untar.ac.id²⁾, H8_KR_Junior@yahoo.com³⁾

ABSTRACT

City Protector is an Arcade Tower Defense Game with endless element. This game implements touch gesture recognizer feature. This game is developed with Unity game engine and playable in Android-based smartphone. To kill zombies, players must draw the symbols contained in the explosives that brought by zombies. Testing is done by blackbox testing, alpha testing, and beta testing through a survey of 32 respondents. The results indicate that City Protector has interesting features and gameplay concepts. It is provides new experiences because the Touch Gesture Recognizer feature is rarely found in general.

Key words

Android, Arcade, City Protector, Endless, Touch Gesture Recognizer, Unity.

1. Pendahuluan

Salah satu program yang dapat berjalan di dalam perangkat berbasis komputer adalah program *game* atau program permainan. *Game* dapat dengan mudah dijalankan pada komputer atau smartphone.[1]

Pengertian dari *game* itu sendiri adalah sebuah aktivitas atau kontes secara fisik ataupun mental yang memiliki sebuah aturan dan dilakukan oleh orang-orang untuk kesenangannya.[2] Sedangkan pengertian dari *game arcade* yaitu *game* yang tidak terfokus pada cerita, melainkan hanya dimainkan untuk bersenang-senang sebagai pengisi waktu senggang atau hanya untuk mencari nilai tertinggi saja.[3]

Game arcade sangat cocok untuk digunakan sebagai sarana untuk melatih kemampuan koordinasi tubuh. Tidak hanya berguna bagi anak-anak saja, melainkan untuk orang dewasa maupun lansia. Interaksi antara mata, otak, serta tangan dapat membantu menyeimbangkan kerja sama antara anggota tubuh. Tentu saja hal ini dapat membuat kecepatan berpikir, melihat, serta bergerak menjadi lebih cepat dan terarah.[4] Fokus *game arcade* adalah refleks pengguna

dalam menekan tombol dan tidak terlalu banyak mengandung unsur lain seperti puzzle atau strategi.[5]

Game yang dirancang memiliki judul *City Protector*, yaitu *game* dengan genre Arcade Tower Defense dengan unsur endless yang pemainnya dapat menggambar dan membentuk *gesture* pada layar Android menggunakan touch gesture recognize untuk mempertahankan kota dari serangan gerombolan *zombie* yang datang untuk menghancurkan kota dengan membawa peledak. Penggunaan *touch gesture* untuk *game arcade* dimaksudkan supaya *game* menjadi lebih menarik dan berbeda dari *game arcade* yang lain. *Game* tersebut dibuat dengan menggunakan Unity dan dimainkan dengan menggunakan platform Android.

Game sejenis yang sebelumnya sudah dibuat adalah *Summoner's Quest* yang merupakan rancangan *game* oleh mahasiswa Universitas Tarumanaga, Fakultas Teknologi Informasi, Teknik Informatika, Bill Kevin (NPM 535120024). *Game* ini adalah *game* yang bergenre Arcade Tower Defense. Pada *game* ini pemain harus menghancurkan base musuh dengan menggambar simbol pada layar untuk mengeluarkan prajurit yang akan mengalahkan monster-monster musuh yang melindungi basemusuh.[6] Contoh tampilan dari *game* dapat dilihat pada Gambar 1.



Gambar 1 Summoner's Quest

2. Dasar Teori

Terdapat beberapa hal yang harus diperhatikan dalam perancangan *game* agar dapat menghasilkan *game* dengan kualitas yang baik. Hal-hal yang perlu dijelaskan adalah perancangan, genre *game*, *control design*, *game engine* yang digunakan dan beberapa hal lainnya yang berhubungan dengan perancangan *game* ini.

2.1 Perancangan

Sebelum suatu *game* dibuat, diperlukan metode perancangan yang berguna sebagai patokan dalam proses pembuatan *game* dan juga menentukan lingkup dari *game* yang ingin dibuat. Terdapat beberapa hal yang harus diperhatikan yaitu sebagai berikut:[7]

1. High Concept
High Concept merupakan deskripsi singkat dari *game City Protector*. *City Protector* merupakan *game* Android yang memiliki genre *Arcade Tower Defense*
2. Gameplay
Gameplay menjelaskan tentang apa yang dapat dilakukan pemain dalam *game City Protector*. Pada *game* ini pemain harus mempertahankan kota dari *zombie* yang membawa peledak. Pemain menggunakan *gesture* untuk menggambar simbol yang sesuai dengan peledak yang dibawa oleh *zombie* pada layar. *Zombie* akan mati jika simbol yang digambar sesuai dengan peledak yang dibawanya.
3. Audience
Audience berisikan tentang sasaran *game City Protector* ditujukan. *Game City Protector* ditujukan bagi pemain dengan usia minimal 10 tahun keatas atau paling tidak mengerti penggunaan *gesture* karena *gesture* merupakan unsur utama untuk memainkan *game* ini.
4. Perangkat Keras
Spesifikasi perangkat keras yang digunakan untuk pembuatan *game* ini adalah prosesor Intel® Core™ i3-3217U, Ram 4GB, dan menggunakan NVIDIA GeForce 720M.
5. Pembuatan Game
Perangkat lunak yang digunakan untuk membuat *game* ini adalah Unity Game Engine dan bahasa pemrograman C#. Untuk membuat komponen grafis dalam *game* digunakan perangkat lunak Adobe Photoshop dan Aseprite.
6. Testing
Setelah *game* selesai dibuat, maka perlu dilakukan tahap *testing* untuk memastikan *game* yang dibuat sudah berjalan dengan baik. Pengujian *game City Protector* dilakukan dengan menggunakan metode Blackbox Testing, Alpha Testing, dan Beta Testing.

2.2 Genre Game

Genre dari *game City protector* adalah *arcade tower defence* yang bersifat *endless*.

Arcade game merupakan jenis *game* yang mengandalkan lebih banyak refleks pemain dalam memegang kontrol dan membutuhkan sedikit pemikiran kompleks. *Arcade games* mempunyai waktu bermain yang singkat, kontrol yang tidak rumit, dan tingkat kesulitan yang terus bertambah.[8]

Genre *strategy* secara umum fokus pada kemampuan berpikir pemain dalam membuat strategi atau taktik yang tepat untuk memenangkan *game*. Tower Defense adalah bagian dari genre *real-time strategy game*. Pemain harus melindungi markas atau base pemain dari musuh-musuh yang datang melalui suatu jalur yang telah ditentukan. [9] Contoh *game* dari sub-genre ini adalah Summoner's Quest.

2.3 Control Design

Setiap *game* memerlukan sebuah perantara yang yang disebut sebagai *game controller* untuk menciptakan interaksi didalamnya. *Game controller* merupakan sebuah perangkat yang digunakan dalam memberikan perintah ke dalam suatu *game*. Keberadaan sebuah *game controller* sangat penting dalam sebuah *game*, karena *game controller* memberikan perintah untuk menggerakkan suatu objek, mengeluarkan senjata, dan hal-hal lainnya yang ada didalam sebuah *game*. Biasanya *game controller* dirancang sedemikian rupa untuk memudahkan pemain dalam melakukan interaksi di dalam *game*. [10]

Game controller yang digunakan dalam *game City Protector* adalah *touch screen* pada perangkat Android untuk menggambar *gesture* yang akan digunakan dalam *game*. Penggunaan *touch screen* sangat mudah karena pemain hanya perlu menyentuh layar dari perangkat yang dimilikinya untuk melakukan interaksi. [11]

2.4 Unity

Unity merupakan *game engine* yang dipakai untuk membuat *game City Protector*. Unity memiliki asset *Point-Cloud Gesture Recognizer* yang digunakan sebagai *recognizer gesture* yang merupakan kelebihan utama pada *game City Protector*.

2.5 Touch Screen Gesture

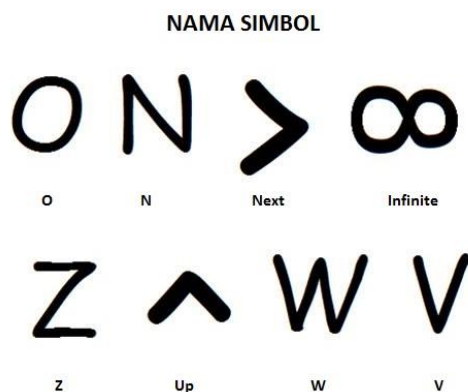
Gestures dibagi menjadi Mekanika Sentuh dan Aktivitas Sentuh. Mekanika sentuh memiliki beberapa hasil tergantung konteks. Aktivitas sentuh dapat dicapai melalui beberapa mekanik sentuh. [12] *Game City Protector* menggunakan Touch, Drag, Swipe, dan Fling sebagai mekanika sentuhnya.

2.6 Touch Gesture Recognition

Gesture Recognition merupakan teknik untuk mengidentifikasi bentuk spesial yang bukan merupakan handwriting, melainkan sebagai indikator dari suatu perintah khusus.[13] Metode gesture recognition yang akan digunakan berupa pointcloud recognizer. Point-cloud recognizer ini akan mengubah bentuk input gesture menjadi kumpulan titik atau points. Point-cloud recognizer ini hanya akan menerima input dari jari pertama yang menyentuh layar. Saat pemain menyentuh jari pada layar, point-cloud recognizer ini akan membaca input pemain setiap frame-nya sampai pemain berhenti menyentuh layar. Input ini akan dibaca sebagai titik-titik atau points dengan koordinat X dan Y. Koordinat points ini akan dibandingkan dengan koordinat points yang terdapat pada files yang telah disimpan sebelumnya. Jika gambar input pemain berbeda ukuran dengan gambar yang terdapat pada files, ukuran gambar tersebut akan diubah skalanya dengan proses normalisasi. Point input dan point yang terdapat pada files akan mengalami proses resampling agar jumlah point input dan yang terdapat pada file memiliki jumlah yang sama. Gesture recognition ini digunakan untuk menggambarkan suatu simbol atau bentuk tertentu pada layar sebagai input untuk mengalahkan musuh.

3. Alur Aplikasi

Game City Protector adalah *game* ber-genre Arcade Tower Defense. Dalam *game* ini pemain ditugaskan untuk melindungi kota dari serangan *zombie* yang membawa peledak. *Zombie* akan berjalan menuju gerbang kota dan akan meledakkan diri di gerbang kota apabila tidak dibunuh. Pemain harus menggambar simbol yang terdapat pada peledak yang dibawa *zombie*. *Zombie* akan mati bila simbol yang digambar pada layar *touch screen* sesuai dengan peledak yang dibawanya. Simbol-simbol tersebut dapat dilihat pada **Gambar 2**. Terdapat 4 jenis *zombie* yang harus dikalahkan oleh pemain. Tampilan *zombie* dapat dilihat pada **Gambar 3**. Status setiap *zombie* dapat dilihat pada **Tabel 1**.



Gambar 2 Simbol Pada Peledak



Gambar 3 Tampilan Zombie

Tabel 1 Status Zombie

Nama Zombie	Movement Speed	Peledak yang Dibawa
Fatty	1	1
Old Man	1	2
Bandana	1.25	2
Soldier	0.75	3

4. Hasil Pengujian

Pengujian *game* dilakukan untuk memastikan bahwa *game* telah dapat berjalan dengan baik. Pengujian *game City Protector* dilakukan dengan 3 metode, yaitu *blackbox testing*, *alpha testing* dan *beta testing*.

4.1 Blackbox Testing

Pengujian dengan metode *blackbox testing* yang dilakukan pada *game City Protector* berupa pengujian terhadap modul untuk memastikan apakah setiap modul pada *game* sudah berjalan dengan baik. Modul-modul yang diuji pada *blackbox testing* adalah sebagai berikut:

1. Modul Home

Pada modul *Home* terdapat 3 buah tombol yaitu *play*, *tutorial*, dan *exit*. Tombol *play* berfungsi untuk berpindah ke modul *In-Game*. Tombol *tutorial* berfungsi untuk berpindah ke modul *Tutorial*. Tombol *exit* berfungsi untuk keluar dari *game*. Tampilan modul *home* dapat dilihat pada **Gambar 4**.

2. Modul Credit

Pada modul ini ditampilkan tentang informasi pembuat *game*, dosen pembimbing, sumber dari asset suara yang digunakan, dan tombol *home*. Tombol *home* berfungsi untuk kembali ke modul *home*. Tampilan modul *Credit* dapat dilihat pada **Gambar 5**.

3. Modul Tutorial

Pada modul ini menampilkan tentang informasi cara memainkan *game*, tombol *home* dan tombol *play*. Tombol *home* berfungsi untuk kembali ke modul *Home* dan tombol *play* berfungsi untuk menuju ke modul *In-Game*. Tampilan modul *Tutorial* dapat dilihat pada **Gambar 6**.

4. Modul In-Game

Pada modul ini terdapat informasi *Score*, *Health Point* dari Gerbang Kota, dan tombol *pause*. Tombol

pause berfungsi untuk memunculkan modul *Pause* dan menghentikan *game* sementara. Tampilan modul *In-Game* dapat dilihat pada **Gambar 7**. Touch Gesture Recognizer pada *game* ini sudah berfungsi dengan baik, *zombie* yang mati sudah sesuai dengan simbol yang digambar.

5. Modul Pause

Pada saat modul ini muncul, *game* akan dihentikan sementara. Pada modul ini terdapat tombol *resume* dan *main menu*. Tombol *resume* berfungsi untuk melanjutkan *game* dan tombol *main menu* berfungsi untuk kembali ke modul *home*. Tampilan modul *Pause* dapat dilihat pada **Gambar 8**.

6. Modul Result

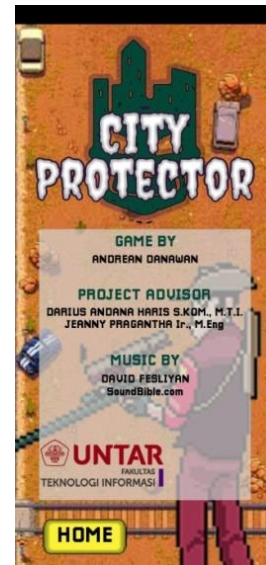
Pada saat modul ini terdapat informasi:

- Informasi *score game* yang sudah dijalankan
- Informasi *highscore* yang atau skor tertinggi yang pernah dicapai
- tombol *play again* yang berfungsi untuk kembali memainkan *game* pada modul *In-Game*
- tombol *home* yang berfungsi untuk kembali ke modul *home*.

Tampilan modul *Pause* dapat dilihat pada **Gambar 9**.



Gambar 4 Tampilan Modul Home



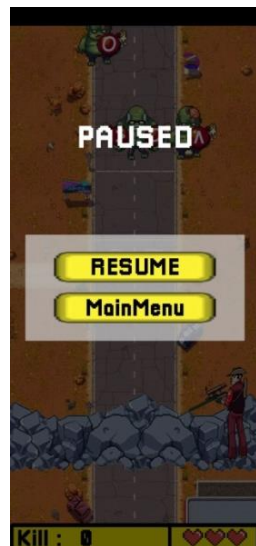
Gambar 5 Tampilan Modul Credit



Gambar 6 Tampilan Modul Tutorial



Gambar 7 Tampilan Modul In-Game



Gambar 8 Tampilan Modul Pause



Gambar 9 Tampilan Modul Result

Berdasarkan pengujian diatas, didapatkan bahwa semua modul telah berjalan dengan baik dan tidak terdapat kesalahan fungsi tombol serta informasi yang ditampilkan.

4.2 Alpha Testing

Alpha testing dilakukan oleh pihak yang mengerti bidang pemrograman. *Alpha testing* yang dilakukan pada game *City Protector* ini dilakukan oleh dosen pembimbing, sebagai pihak yang telah mengerti konsep dari game ini. Penguji memberikan masukan untuk mengganti tulisan dan tombol menjadi bergaya pixel, komentar mengenai simbol W terkadang tidak dapat dikenali, dan belum ada gambar icon game.

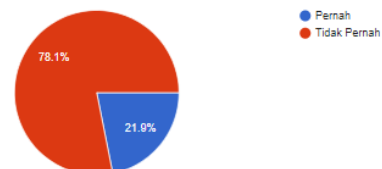
Dengan dilakukannya *alpha testing*, dapat diketahui bahwa masih terdapat kekurangan yang harus diperbaiki pada game *City Protector* ini. Komentar, dan masukan kekurangan yang ditemukan pada saat *alpha testing* telah diperbaiki sesuai dengan petunjuk dari penguji agar game dapat berjalan dengan lebih baik.

4.3 Beta Testing

Beta testing dilakukan dengan membiarkan penguji memainkan game *City Protector*. Pengujian ini dilakukan oleh 32 responden pada tanggal 7 Juni 2020 sampai 13 Juni 2020. Pengujian dilakukan dengan menyebarkan kuesioner pada Goole Form untuk diisi secara langsung setelah memainkan game *City Protector* menggunakan *smartphone* yang telah disediakan oleh penguji kepada 24 orang responden dan menyebarkan kuesioner pada Goole Form untuk diisi setelah game *City Protector* diunduh dan dimainkan kepada 8 orang. Kuesioner berisi 8 pertanyaan yang terdiri dari pengetahuan mengenai game serupa, dan pendapat mengenai game *City Protector*. Contoh hasil dari jawaban responden dapat dilihat pada **Gambar 10** dan **Gambar 11**.

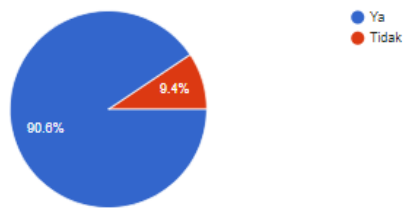
Apakah anda pernah memainkan permainan dengan fitur menggambar simbol sebagai input sebelumnya?

32 responses



Gambar 10 Game Dengan Fitur Menggambar Simbol

Apakah zombie yang mati sudah sesuai dengan simbol yang anda gambar?
32 responses



Gambar 11 Kesesuaian Simbol yang Digambar Dengan Zombie yang Mati

4.4 Pembahasan Hasil Pengujian

Berdasarkan jawaban yang telah diberikan oleh 32 responden, terkumpul hasil sebagai berikut:

1. Sebanyak 78,10% responden menyatakan tidak pernah memainkan *game* dengan fitur menggambar simbol sebagai *input* dan sebanyak 87,50% responden menyatakan konsep menggambar simbol untuk mengalahkan musuh memberikan pengalaman yang baru, sehingga dapat disimpulkan fitur menggambar simbol dalam *game City Protector* masih sedikit ditemukan pada *game* pada umumnya sehingga memberikan pengalaman yang baru bagi pemain.
2. Semua responden menyatakan simbol pada peledak yang dibawa *zombie* cukup jelas untuk digambar.
3. 90,6% responden menyatakan *zombie* yang mati sudah sesuai dengan simbol yang digambar. Hal ini disebabkan karena responden tidak dapat menggambar simbol yang sesuai. Sebagai contoh simbol N digambar mirip dengan simbol W sehingga *touch gesture recognizer* salah mengenali simbol.
4. Dari 8 macam simbol, sebanyak 46,90% responden menyatakan simbol berbentuk next merupakan simbol pada peledak yang paling mudah digambar.
5. Dari 8 macam simbol, sebanyak 65,60% responden menyatakan simbol berbentuk infinite merupakan simbol pada peledak yang paling sulit digambar.
6. Sebanyak 84,40% responden menyatakan konsep menggambar simbol sebagai *gameplay* menarik dan 88,89% diantara responden tersebut tertarik untuk memainkan *game City Protector* lagi, sehingga dapat disimpulkan *game City Protector* merupakan *game* yang menarik sehingga pemain tertarik untuk memainkan lagi.

5. Kesimpulan dan Saran

Setelah pengujian terhadap *game City Protector* selesai dilakukan, kesimpulan yang dapat ditarik dari hasil pengujian tersebut adalah sebagai berikut:

1. Fitur menggambar simbol menggunakan *Touch Gesture Recognizer* dalam *game City Protector*

masih jarang ditemukan pada *game* umumnya sehingga memberikan pengalaman yang baru bagi pemain.

2. Konsep *gameplay* *City Protector* dianggap menarik sehingga pemain tertarik untuk memainkan lagi.
3. Tampilan peledak dalam *game City Protector* sudah cukup jelas untuk digambar pemain.
4. Sebagian besar *zombie* yang mati sudah sesuai dengan simbol yang digambar, maka *training touch gesture recognizer* dirasa cukup dan berfungsi dengan baik.
5. *Touch Gesture Recognizer* ini hanya menerima input dari jari pertama yang menyentuh layar, sehingga tidak mungkin untuk menggambar 2 simbol secara bersamaan.

Selain kesimpulan yang diambil dari data dan komentar yang diterima saat pengujian, terdapat juga saran-saran yang dapat digunakan untuk mengembangkan *game* ini. Berikut ini adalah saran-saran yang disampaikan:

1. Jenis *zombie* dibuat lebih beragam dan memiliki kemampuan unik supaya membuat *game* lebih menarik misalnya *zombie* dengan efek racun yang memiliki kemampuan membuat layar *smartphone* gelap sesaat waktu dibunuh.
2. Untuk pengembangan *game* ini hindari penggunaan simbol yang memiliki bentuk mirip satu sama lain untuk mempermudah proses *training* dan menghindari kesalahan pada saat dilakukan *recognize gesture*.
3. Untuk pengembangan *game* ini antisipasi *recognize gesture* yang keliru dikarenakan gaya gambar simbol dari tiap pemain berbeda dengan melakukan *training gesture* dengan gaya gambar tiap simbol yang lebih beragam.

REFERENSI

- [1] Mandalamaya. Pengertian Game Menurut Para Ahli. <http://www.mandalamaya.com/pengertian-game-menurut-para-ahli/>, 1 September 2019.
- [2] Merriam-Webster. Game | Definition of game by Merriam-Webster. <http://www.merriam-webster.com/dictionary/game>, 12 Agustus 2019.
- [3] KapanLagi. 10 Kategori Game dan Manfaatnya. <https://game.kapanlagi.com/berita/10-kategori-game-dan-manfaatnya.html>, 1 September 2019.
- [4] Anita. Pengertian dan Sejarah Game Arcade Lengkap. <https://www.daftarinformasi.com/permainan-game-arcade/>, 1 September 2019.
- [5] Silvianita, Helva. Pengertian Game Beserta Sejarah, Manfaat, serta Jenis-Jenis Game, Lengkap. <https://www.nesabamedia.com/pengertian-game/>, 1 September 2019.
- [6] Kevin, Bill. 2016. PEMBUATAN GAME TOWER DEFENSE SUMMONER'S QUEST MENGGUNAKAN UNITY DENGAN TOUCH GESTURE RECOGNIZER. Jurnal Ilmu Komputer dan Sistem Informasi. Vol.4 No.2
- [7] Bates, Bob. Game Design Second Edition. (Stamford: Cengage Learning PTR, 2004). h. 204
- [8] Technopedia. Arcade Game. <http://www.technopedia.com/definition/1903/arcade-game>, 22 Februari 2020.

- [9] Wikipedia. List of video game genres. https://en.wikipedia.org/wiki/List_of_video_game_genres, 24 Februari 2020.
- [10] Sethi, Maneesh. Game Programming for Teens Second Edition. (Boston: Course Technology PTR, 2005), h.277.
- [11] DarkWatcher. Game Controller. <http://www.videogameconsolelibrary.com/art-controller.htm#page=motion>, 24 Februari 2020.
- [12] Google. Gestures. <https://material.io/design/interaction/gestures.html>. 20 Februari 2020.
- [13] Liquiseach. Pen Computing – General Techniques of Pen Computing – Gesture Recognition. http://www.liquiseach.com/pen_computing/general_techniques_of_pen_computing/gesture_recognition. 25 September 2019

Andrean Danawan, mahasiswa S1, program studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara.

Ir. Jeanny Pragantha, M.Eng, memperoleh Ir dari institute Teknologi Bandung. Kemudian memperoleh gelar M. Eng. Dari Asian Institut of Technology, Bangkok. Saat ini sebagai dosen program studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara.

Darius Andana Haris, M.TI, memperoleh gelar S.Kom dari Universitas Tarumanagara pada 2009, melanjutkan S2 di Universitas Bina Nusantara dan memperoleh gelar M.TI. Saat ini sebagai dosen program studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Tarumanagara.