

LAPORAN TUGAS BESAR
IF1210 DASAR PEMROGRAMAN
SIMULASI PROGRAM GAME BINOMO

Disusun oleh:

Kelas K-08
Kelompok 10

Satria Octavianus Nababan	16521034
Fawwaz Abrial Saffa	16521115
Frendy Sanusi	16521124
Adinda Siti Fathonah	16521160



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

LEMBAR PERNYATAAN

“Saya menyatakan bahwa saya mengerjakan tugas besar ini dengan sejujur-jujurnya, tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui saya mengerjakan tugas besar ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2020/2021.”

Kami yang menyatakan:

- | | |
|-----------------------------|----------|
| 1. Satria Octavinus Nababan | 16521034 |
| 2. Fawwaz Abrial Saffa | 16521115 |
| 3. Frendy Sanusi | 16521124 |
| 4. Adinda Siti Fathonah | 16521160 |

DAFTAR ISI

LEMBAR PERNYATAAN	1
DAFTAR ISI	2
DAFTAR TABEL	3
DAFTAR GAMBAR	4
1. DESKRIPSI PERSOALAN	6
2. PEMBAGIAN TUGAS.....	8
3. CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING SETIAP PRIMITIF	10
4. DESAIN COMMAND UNTUK SETIAP PRIMITIF	11
5. DESAIN KAMUS DATA.....	14
5. DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM.....	18
6. SPESIFIKASI SETIAP MODUL/PROSEDUR/FUNGSI DALAM NOTASI ALGORITMIK....	20
7. SCREENSHOT HASIL PENGUJIAN PROGRAM.....	47
LAMPIRAN	52

DAFTAR TABEL

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok

Tabel 2. Tabel Checklist Hasil Rancangan, Implementasi, dan Testing

DESAIN KAMUS DATA

Tabel 3. Deklarasi Type

Tabel 4. Deklarasi Fungsi dan Prosedur account.py

Tabel 5. Deklarasi Fungsi dan Prosedur showgame.py

Tabel 6. Deklarasi Fungsi dan Prosedur modifygame.py

Tabel 7. Deklarasi Fungsi dan Prosedur system.py

Tabel 8. Deklarasi Fungsi dan Prosedur help.py

Tabel 9. Deklarasi Fungsi dan Prosedur menu.py

Tabel 10. Deklarasi Fungsi dan Prosedur misc.py

SPESIFIKASI SETIAP MODUL/PROSEDUR/FUNGSI DALAM NOTASI ALGORITMIK

Tabel 11. account.py

Tabel 12. showgame.py

Tabel 13. modifygame.py

Tabel 14. system.py

Tabel 15. help.py

Tabel 16. menu.py

DAFTAR GAMBAR

DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM

Gambar 1. Flowchart Program Sebagai Admin + Admin dan User.

Gambar 2. Flowchart Program Sebagai User

SCREENSHOT HASIL PENGUJIAN PROGRAM

Gambar 3. Pengujian Kerja Fungsi Load (1)

Gambar 4. Pengujian Fungsi Load (2)

Gambar 5. Menu Login Awal

Gambar 6. Pengujian Fungsi Login

Gambar 7. Menu Program Utama

Gambar 8. Kolom Pengisian Fungsi Register

Gambar 9. Testing Fungsi Register (1)

Gambar 10. Testing Fungsi Register (2)

Gambar 11. Testing Fungsi Register (3)

Gambar 12. Kolom Pengisian Fungsi Menambah Game

Gambar 13. Testing Fungsi Menambah Game (1)

Gambar 14. Testing Fungsi Menambah Game (2)

Gambar 15. Kolom Pengisian Fungsi Mengubah Game

Gambar 16. Testing Fungsi Mengubah Game (1)

Gambar 17. Testing Fungsi Mengubah Game (2)

Gambar 18. Testing Fungsi Mengubah Game (3)

Gambar 19. Kolom Pengisian Fungsi List Game Toko

Gambar 20. Testing Fungsi List Game Toko (1)

Gambar 21. Testing Fungsi List Game Toko (2)

Gambar 22. Testing Fungsi List Game Toko (3)

Gambar 23. Testing Fungsi List Game Toko (4)

Gambar 24. Testing Fungsi List Game Toko (4)

Gambar 25. Kolom Pengisian Fungsi Cari Game Toko

Gambar 26. Testing Fungsi Cari Game Toko (1)

Gambar 27. Testing Fungsi Cari Game Toko (2)

Gambar 28. Testing Fungsi Cari Game Toko (3)

Gambar 29. Kolom Pengisian dan Testing Fungsi Topup (1)

Gambar 30. Testing Fungsi Topup (2)

Gambar 31. Testing Fungsi Topup (3)

Gambar 32. Testing Fungsi Topup (4)

Gambar 33. Testing Fungsi Help Untuk Admin

Gambar 34. Testing Fungsi Help Untuk User

Gambar 35. Kolom Pengisian dan Testing Fungsi Save
Gambar 36. Testing Fungsi Save (1)
Gambar 37. Testing Fungsi Save (2)
Gambar 38. Kolom Pengisian Fungsi Exit
Gambar 39. Testing Fungsi Exit (1)
Gambar 40. Testing Fungsi Exit (2)
Gambar 41. Tampilan Hasil Fungsi Riwayat
Gambar 42. Kolom Pengisian Fungsi Beli Game
Gambar 43. Testing Fungsi Beli Game (1)
Gambar 44. Testing Fungsi Beli Game (2)
Gambar 45. Testing Fungsi Beli Game (3)
Gambar 46. Testing Fungsi Beli Game (4)
Gambar 47. Tampilan Hasil Fungsi List Game
Gambar 48. Kolom Pengisian Fungsi Cari Game Saya
Gambar 49. Testing Fungsi Cari Game Saya (1)
Gambar 50. Testing Fungsi Cari Game Saya (2)
Gambar 51. Testing Fungsi Cari Game Saya (3)
Gambar 52. Kolom Pengisian Fungsi Magic Conch Shell
Gambar 53. Testing Fungsi Fungsi Magic Conch Shell (1)
Gambar 54. Testing Fungsi Fungsi Magic Conch Shell (2)
Gambar 55. Tampilan Permainan Fungsi Tic Tac Toe
Gambar 56. Testing Permainan Fungsi Tic Tac Toe (1)
Gambar 57. Testing Permainan Fungsi Tic Tac Toe (2)
Gambar 58. Testing Permainan Fungsi Tic Tac Toe (3)
Gambar 59. Testing Permainan Fungsi Tic Tac Toe (4)
Gambar 60. Testing Permainan Fungsi Tic Tac Toe (5)

LAMPIRAN

Gambar 61. Form Asistensi 1
Gambar 62. Form Asistensi 2
Gambar 63. Dokumentasi

1. DESKRIPSI PERSOALAN

Program ini bertujuan untuk memperbaiki BNMO, BNMO (dibaca: Binomo) adalah sebuah robot *game* milik Indra dan Doni yang membantu mereka melepas stress ketika mendapatkan tugas selama di Institut Teknologi Bandung. BNMO dulunya memiliki sistem inventarisasi & toko game yang baik. Indra dan Doni menjalani kuliah 2 semester di ITB dan merasa kesulitan dan stress. Doni menghabiskan waktu dengan BNMO untuk bermain *game*. Namun, Indra lebih suka bersenang-senang bermain *gacha*, akan tetapi ia rugi terus. Sehingga, pada suatu saat, Indra membanting BNMO sehingga BNMO pun rusak. Doni merasa depresi saat ia tahu BNMO rusak.

Saat memainkan BNMO admin dapat mendaftarkan pengguna baru dengan memasukkan nama, username, dan password. Pengguna yang mendaftar otomatis memiliki role “user”. Pastikan username bersifat unik. Fungsi ini tidak dapat membuat user dengan role admin; untuk membuat user admin dapat mengedit file penyimpanan. Username hanya dapat mengandung alfabet A-Z dan a-z, *underscore* “_”, *strip* “-”, dan angka 0-9. Jika *user* sudah pernah melakukan registrasi maka user dapat langsung melakukan login ke *game*, saat masuk ke dalam aplikasi, pengguna bisa login dengan memasukkan *username* dan *password*. Bila *username* dan *password* yang diinput cocok dengan username dan password pada file user, maka pengguna berhasil login. Jika username dan password yang diinput tidak cocok, maka tidak dapat login dan pengguna diminta memasukkan username dan password yang benar.

Admin dapat menambahkan game ke toko, penambahan game pada toko game dilakukan melalui pengisian informasi game yang akan ditambahkan. Program lalu akan melakukan validasi apakah semua informasi yang dibutuhkan telah *diinput* oleh *admin*. Apabila terdapat informasi yang belum dimasukkan oleh pengguna, program akan meminta *input* lagi kepada pengguna. Hal ini akan dilakukan terus menerus sampai pengguna telah melakukan *input* semua informasi game. Game baru yang telah ditambahkan akan disimpan pada file csv. Selain menambahkan game, admin juga dapat mengubah game pada toko. Mengubah game pada toko game dilakukan melalui pengisian informasi game yang akan diubah. Setelah itu game dengan informasi baru yang telah diubah akan disimpan pada file csv. Admin juga tidak harus mengisi semua field, sehingga saat admin ingin membiarkan value field tertentu agar sesuai dengan value field data yang lama, admin dapat melewatinya dengan mengosongkan. Tidak bisa mengubah stock pada fungsional ini. Mengubah stok sebuah game pada toko dilakukan melalui input ID dan besar perubahan stok yang ingin dilakukan. Saat dilakukan pengubahan stok suatu game, perlu dilakukan validasi untuk memastikan stok game tersebut tetap valid setelah pengubahan (tidak negatif). Bila stok suatu game bernilai nol setelah pengubahan, tidak perlu dihapus dari sistem.

Selanjutnya yaitu membuat fitur listing game di toko berdasarkan ID, Tahun Rilis dan Harga. Agar pengguna dapat melihat *game-game* yang ada di toko, toko harus bisa memberikan daftar *game* yang dimiliki. Beberapa pengguna mungkin merupakan *gamer* yang *up-to-date* terhadap perkembangan game, sehingga ingin melihat *game* yang terbaru, sedangkan beberapa di antaranya merupakan *gamer* yang suka mengoleksi *game* klasik. Selain itu juga ada pengguna yang merupakan *gamer* dengan budget pas-pasan, sehingga perlu berpikir dua kali ketika membeli *Fifa 22*. Oleh karena itu, selain menyediakan daftar *game*, toko juga harus bisa melakukan *sorting* terhadap *game* berdasarkan harga dan tahun rilisnya. Pengguna bisa memberikan inputan berupa skema *sorting* dari daftar *game* yang ingin dilihat ke dalam aplikasi. Terdapat 2 skema *sorting*,

yaitu berdasarkan tahun rilis atau harga (satu saja, tidak ada kasus di sort berdasarkan dua attribut), urutan bisa *ascending* atau *descending*. Pastikan skema sorting tervalidasi. Parsing input skema sorting dibebaskan. Jika skema sorting dikosongkan, akan di sort berdasarkan ID *ascending*.

Terdapat juga fitur untuk membeli game., pada fitur ini Game yang telah dibeli akan masuk ke list Game yang dimiliki User. Game hanya dapat dibeli user yang sama sebanyak satu kali. Terdapat 1 parameter yang wajib diisi pada prosedur ini, yaitu ID Game yang akan dibeli user. Fitur untuk mencari game yang dimiliki dari ID dan tahun rilis, Prosedur ini digunakan untuk mendapatkan informasi game sesuai dengan query yang diminta oleh pengguna pada inventory. Terdapat 2 parameter yang dapat digunakan, yaitu ID Game dan Tahun Rilis Game. Parameter bersifat tidak wajib diisi. Jadi, apabila user ingin mencari game dengan kategori tertentu saja, ia dapat mengosongkan parameter lain dengan tidak mengisi input apapun. Ketika semua parameter dikosongkan, maka sistem akan menampilkan semua list game yang dimiliki oleh user tersebut. Pengguna juga dapat mencari game di toko dari ID, Nama Game, Harga, Kategori dan Tahun Rilis. Seperti fungsional sebelumnya, namun kali ini pencarian dilakukan pada toko. Terdapat 5 parameter yang dapat digunakan, yaitu ID Game, Nama Game, Harga, Kategori dan Tahun Rilis Game. Parameter bersifat tidak wajib diisi.

Pengguna juga dapat melakukan top up saldo melalui fitur top up saldo untuk menambahkan saldo kepada User. Masukan menerima sebuah username dan sejumlah saldo. Saldo ini kemudian ditambahkan ke User dengan username yang sesuai. Username perlu dilakukan validasi. Saldo dapat bernilai minus untuk mengurangi. Namun, apabila masukan saldo negatif, perlu dilakukan validasi. Setelah melakukan banyak pembelian *game*, pengguna bisa saja ingin melihat riwayat pembelian *game* karena ingin mengetahui berapa harga beli dan/atau kapan pengguna membeli *game-game* yang sudah dimiliki. Fitur ini akan menampilkan pesan khusus ketika pengguna tidak memiliki riwayat pembelian *game*. Untuk memberikan panduan penggunaan sistem, pengguna dapat menggunakan fitur help. Keluaran memberikan arahan yang cukup jelas dengan akses yang bersesuaian, tidak perlu melakukan login dulu untuk menggunakan menu ini.

Terdapat fitur load yang digunakan untuk melakukan loading data ke dalam sistem. Prosedur ini akan otomatis dijalankan ketika sistem mulai pertama kali bila diberikan input nama folder yang berisi file penyimpanan. Semua file penyimpanan dalam suatu folder dijamin ada dan memiliki nama yang fixed, seperti yang tertera pada bagian struktur data eksternal. Namun untuk folder, harus dilakukan validasi. Masukan nama direktori ketika hendak menjalankan script lewat argumen. Cara membaca argumen dari Python menggunakan `argparse`. Selanjutnya membuat fitur save untuk melakukan penyimpanan data ke dalam file setelah dilakukan perubahan. Misalnya, setelah melakukan pembelian atau penjualan tanpa di-*save*, data tersebut tidak akan tersimpan pada file. Beberapa aturan terkait save data:

- Bila nama folder belum ada, buat folder dan simpan file csv ke dalam folder tersebut. Sebaliknya, bila nama folder sudah ada, tidak perlu menghapus folder yang ada.
- Bila nama file sudah ada, hapus file tersebut dan ganti dengan yang baru(*replace/overwrite*). Sebaliknya, buat file baru dan simpan ke file tersebut.

Terakhir, untuk keluar dari aplikasi menggunakan fitur exit, fitur ini akan menanyakan apakah pengguna akan melakukan penyimpanan terhadap file yang diubah atau tidak. Fungsi dapat menerima

huruf kecil maupun besar. Pastikan masukan valid. Kalau tidak valid, bisa tanyakan kembali pertanyaannya. Jika valid, maka akan menjalankan masukan user.

2. PEMBAGIAN TUGAS

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F02	procedure register	16521124	16521124	16521034 16521115 16521124 16521160
	function validasi_user			
F03	procedure login	16521124	16521124	
F04	procedure tambah_game	16521160	16521160	
F05	procedure ubah_game	16521160	16521160	
F06	procedure ubah_stok	16521115	16521115	
F07	procedure list_game_toko	16521034 16521124	16521034 16521124	
	procedure Ascending			
	procedure Descending			
	procedure CetakSorting			
F08	procedure buy_game	16521115	16521115	
F09	procedure list_game	16521115	16521115	
F10	procedure search_my_game	16521124	16521124	
	procedure CetakGame			
F11	procedure search_game_at_store	16521124	16521124	
F12	procedure topup	16521160	16521160	
F13	procedure riwayat	16521160	16521160	
F14	procedure help	16521034	16521034	
F15	procedure load	16521115	16521115	
F16	procedure writeToCSV	16521034	16521034	
	procedure save	16521115	16521115	
F17	procedure exit	16521034	16521034	
B01	function encrypt	16521115	16521115	
	function decrypt			
	function egcd			
	Function inverse_modulo			
B02	procedure kerangajaib	16521124	16521124	
B03	procedure tictactoe	16521115 16521124	16521115 16521124	
	procedure cetak_papan	16521124	16521124	
	procedure cek_pemain			
	function validasi_input			
	function win_condition	16521115	16521115	
	function game_end			
Lain-lain	procedure readAllCSV	16521115	16521115	
	procedure menu_awal	16521124	16521124	
	procedure menu	16521124	16521124	
	function cek_role	16521124	16521124	

	procedure change_admin_or_user	16521124	16521124	
	function Length	16521124	16521124	
	function Append	16521115	16521115	
	function split			
	function readCSV			
	function getIndexFromAttribute			
	procedure Clear	16521124	16521124	
	procedure Wait	16521124	16521124	

3. CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING SETIAP PRIMITIF

Tabel 2. Tabel Checklist Hasil Rancangan, Implementasi, dan Testing

Fitur	Desain	Implementasi	Testing
F02	V	V	V
F03	V	V	V
F04	V	V	V
F05	V	V	V
F06	V	V	V
F07	V	V	V
F08	V	V	V
F09	V	V	V
F10	V	V	V
F11	V	V	V
F12	V	V	V
F13	V	V	V
F14	V	V	V
F15	V	V	V
F16	V	V	V
F17	V	V	V
B01	V	V	V
B02	V	V	V
B03	V	V	V

4. DESAIN COMMAND UNTUK SETIAP PRIMITIF

➤ **F02 - register**

Nama command : register()
Masukan : nama, username, password
Keluaran : Berhasil mendaftarkan akun user

➤ **F03 - login**

Nama command : login()
Masukan : username, password
Keluaran : Berhasil login akun

➤ **F04 - tambah_game**

Nama command : tambah_game ()
Masukan : nama game, kategori, tahun rilis, harga, stok
Keluaran : Berhasil menambahkan game

➤ **F05 - ubah_game**

Nama command : ubah_game ()
Masukan : nama game, kategori, tahun rilis, harga
Keluaran : Berhasil mengubah atribut game

➤ **F06 - ubah_stok**

Nama command : ubah_stok()
Masukan : game_id, jumlah_tambahan
Keluaran : Berhasil menambahkan atau mengurangi stok game

➤ **F07 - list_game_toko**

Nama command : list_game_toko (indeks)
Masukan : harga+,harga-,tahun+,tahun-,dan masukan kosong
Keluaran : Berhasil sorting daftar game sesuai masukan

➤ **F08 - buy_game**

Nama command : buy_game(user_id)
Masukan : game_id
Keluaran : Berhasil membeli game

➤ **F09 - list_game**

Nama command : list_game(user_id)
Masukan : -
Keluaran : Mencetak list game yang dimiliki user

➤ **F10 - search_my_game**

Nama command : search_my_game(id_user)
Masukan : ID, tahun_rilis

- Keluaran : Mencetak game user yang dicari berdasarkan masukan parameter
- **F11 - search_game_at_store**
Nama command : search_game_at_store()
Masukan : ID, nama, harga, kategori, tahun_rilis
Keluaran : Mencetak game yang dicari di toko berdasarkan masukan parameter
 - **F12 - topup**
Nama command : topup ()
Masukan : username, saldo
Keluaran : Topup berhasil!
 - **F13 - riwayat**
Nama command : riwayat (user_id)
Masukan : -
Keluaran : menampilkan riwayat pembelian game
 - **F14 - Help**
Nama command : help (role)
Masukan : -
Keluaran : menampilkan panduan penggunaan game sesuai role
 - **F15 - load**
Nama command : python <nama_file> <nama_folder>
Masukan : nama_file, nama_folder
Keluaran : menampilkan panduan penggunaan game sesuai role
 - **F16 - save**
Nama command : save ()
Masukan : loc { nama folder }
Keluaran : Berhasil membuat folder sesuai lokasi dari input user dan didalam folder
 - **F17 - exit**
Nama command : Exit()
Masukan : pilihan
Keluaran : Berhasil keluar dari aplikasi
 - **B02 - Magic Conch Shell**
Nama command : kerangajaib()
Masukan : tanya
Keluaran : Mencetak jawaban ke layar secara acak

➤ **B03 - Game Tic-Tac-Toe**

Nama command : tictactoe()

Masukan : X, Y

Keluaran : Mencetak hasil permainan

➤ **T01 - change_admin_or_user**

Nama command : change_admin_or_user()

Masukan : username, ubah_role

Keluaran : Berhasil mengganti role user

5. DESAIN KAMUS DATA

Tabel 3. Deklarasi Type	
type User :	< id : <u>integer</u> ; username : <u>string</u> ; nama : <u>string</u> ; password : <u>string</u> ; role : <u>string</u> ; saldo : <u>integer</u> >
type Game :	< id : <u>string</u> ; nama : <u>string</u> ; kategori : <u>string</u> ; tahun_rilis : <u>integer</u> ; harga : <u>integer</u> ; stok : <u>integer</u> >
type Riwayat :	< game_id : <u>string</u> ; nama : <u>string</u> ; harga : <u>integer</u> ; user_id : <u>integer</u> ; tahun_beli : <u>integer</u> >
type Kepemilikan :	< game_id : <u>string</u> ; user_id : <u>integer</u> >

Tabel 4. Deklarasi Fungsi dan Prosedur account.py	
F02	procedure register (<u>input</u> nama : <u>string</u> , <u>input</u> username : <u>string</u> , <u>input</u> password : <u>string</u>) { I.S : nama, username, dan password sembarang F.S : Mencetak pesan ke layar apakah berhasil register atau tidak }
	function validasi_user (username : <u>string</u>) -> <u>boolean</u> { Mengembalikan True jika masukan username benar }
F03	function login () -> (<u>boolean</u> , <u>string</u> , <u>string</u> , <u>integer</u>) { Akses masuk ke program jika masukan username dan password benar }
F12	procedure topup (<u>input</u> username : <u>string</u> , <u>input</u> saldo : <u>integer</u>) { I.S : username dan saldo sembarang F.S : mencetak username dan nilai saldo akhir yang tervalidasi }

Tabel 5. Deklarasi Fungsi dan Prosedur showgame.py	
F07	procedure list_game_toko (<u>input</u> pilih_skema : <u>string</u>) { I.S : pilih_skema terdefinisi F.S : menampilkan list game pada toko berdasarkan tahun+, tahun, harga+, harga-, atau kosong (tekan tombol Enter) }

	procedure Ascending (<u>input</u> indeks : <u>integer</u>) { I.S : indeks terdefinisi F.S : mencetak list game di toko dengan terurut membesar }
	procedure Descending (<u>input</u> indeks : <u>integer</u>) { I.S : indeks terdefinisi F.S : mencetak list game di toko dengan terurut mengecil }
	procedure CetakSorting (<u>output</u> count : <u>integer</u>) { I.S : data.game berisi data csv F.S : mencetak data.game ke layar secara terurut }
F09	procedure list_game (<u>input</u> user_id : <u>string</u>) { I.S : user_id sembarang F.S : Mencetak semua game yang dimiliki oleh user dengan user id tersebut ke layar
F10	procedure search_my_game (<u>input</u> id_user : <u>string</u> , <u>input</u> ID : <u>string</u> , <u>input</u> tahun_rilis : <u>string</u>) { I.S : id_user terdefinisi, ID dan tahun_rilis sembarang F.S : Menampilkan hasil pencarian game ke layar berdasarkan ID dan tahun rilis }
	procedure CetakGame (<u>input</u> indeks : <u>integer</u> , <u>input</u> count : <u>integer</u>) { I.S : indeks dan count terdefinisi F.S : Mencetak hasil pencarian ke layar }
F11	procedure search_game_at_store (<u>input</u> ID : <u>string</u> , <u>input</u> nama : <u>string</u> , <u>input</u> harga : <u>string</u> , <u>input</u> kategori : <u>string</u> , <u>input</u> tahun_rilis : <u>string</u>) { I.S : ID, nama, harga, kategori, dan tahun_rilis sembarang F.S : Menampilkan hasil pencarian game ke layar berdasarkan ID, nama, harga, kategori, dan tahun_rilis }
	procedure CetakGame (<u>input</u> indeks : <u>integer</u> , <u>input</u> count : <u>integer</u>) { I.S : indeks dan count terdefinisi F.S : Mencetak hasil pencarian ke layar }
F13	procedure riwayat (<u>input</u> user_id : <u>string</u>) { I.S : user_id terdefinisi F.S : menampilkan riwayat pembelian game dari user_id terkait }

Tabel 6. Deklarasi Fungsi dan Prosedur modifygame.py	
F04	procedure tambah_game (<u>input</u> nama, kategori, tahun_rilis, harga, stok : <u>string</u>) { I.S : nama, kategori, tahun_rilis, harga, stok sembarang F.S : seluruh data atribut game terdefinisi }
F05	procedure ubah_game (<u>input</u> game_id, nama, kategori, tahun_rilis, harga : <u>string</u>) { I.S : game_id, nama, kategori, tahun_rilis, harga sembarang F.S : menampilkan game_id, nama, kategori, tahun_rilis, dan harga yang telah diubah jika masukkan tidak kosong }

F06	procedure ubah_stok (<u>input</u> game_id : <u>string</u> , <u>input</u> jumlah_tambahan : <u>integer</u>) { I.S : game_id dan jumlah_tambahan sembarang F.S : mengupdate stok game dengan game_id input sebanyak tambahan sesuai input jumlah_tambahan kemudian dicetak perubahan / tidak perubahannya }
F08	procedure buy_game (<u>input</u> user_id, game_id : <u>string</u>) { I.S : game_id dan user_id sembarang F.S : menambahkan game dengan game_id sesuai input ke dalam kepemilikan user dengan user_id sesuai input, mengurangi saldo user, dan mengurangi stok game bila berhasil terbeli }

Tabel 7. Deklarasi Fungsi dan Prosedur system.py

F15	procedure load(<u>input</u> nama_folder : <u>string</u>) { I.S : nama folder penyimpanan sembarang F.S : membaca seluruh file csv pada folder dan menyimpannya dalam array }
F16	procedure save(<u>input</u> loc : <u>string</u>) { I.S : Lokasi sembarang, folder belum tentu ada pada lokasi tersebut F.S : Folder terbuat sesuai lokasi dari input user dan di dalam folder }
	procedure writeToCSV(<u>input</u> path : <u>string</u> <u>input</u> data_array : <u>Array</u> of <u>Array</u> of <u>string</u> <u>input</u> data_name : <u>string</u>) { I.S : path, data_array, dan data_name sembarang F.S : Terbentuk file <data_name>.csv yang berisikan data dalam array data_array }
F17	procedure Exit (<u>input</u> pilihan : <u>string</u>) { I.S : pilihan terdefinisi F.S : mengeksekusi pilihan dan keluar dari aplikasi }

Tabel 8. Deklarasi Fungsi dan Prosedur help.py

procedure Help(<u>input</u> admin : <u>boolean</u>) { I.S : Admin terdefinisi sembarang F.S : Menampilkan panduan penggunaan aplikasi }
--

Tabel 9. Deklarasi Fungsi dan Prosedur menu.py

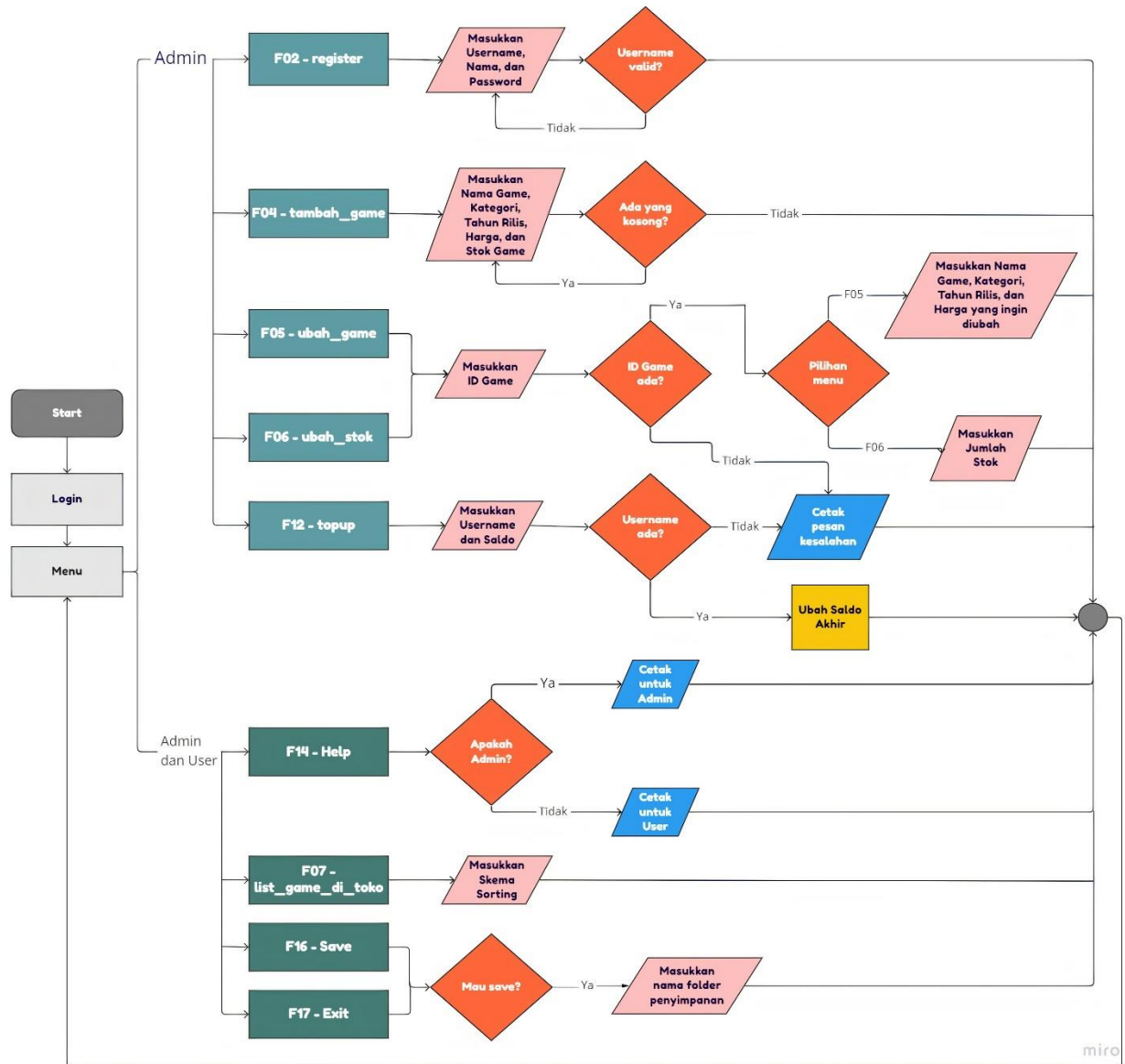
1	procedure menu_awal () { I.S : Menampilkan selamat datang dan memanggil fungsi login F.S : Menampilkan menu baru jika berhasil login }
2	procedure menu (<u>input</u> pilihan : <u>string</u>) { I.S : Pilihan sembarang dan menampilkan menu pilihan ke layar F.S : Memanggil fungsi sesuai masukan }
3	function cek_role (username : <u>string</u>) -> <u>boolean</u> { Mengembalikan nilai True jika role pengguna adalah admin }

Tabel 10. Deklarasi Fungsi dan Prosedur misc.py

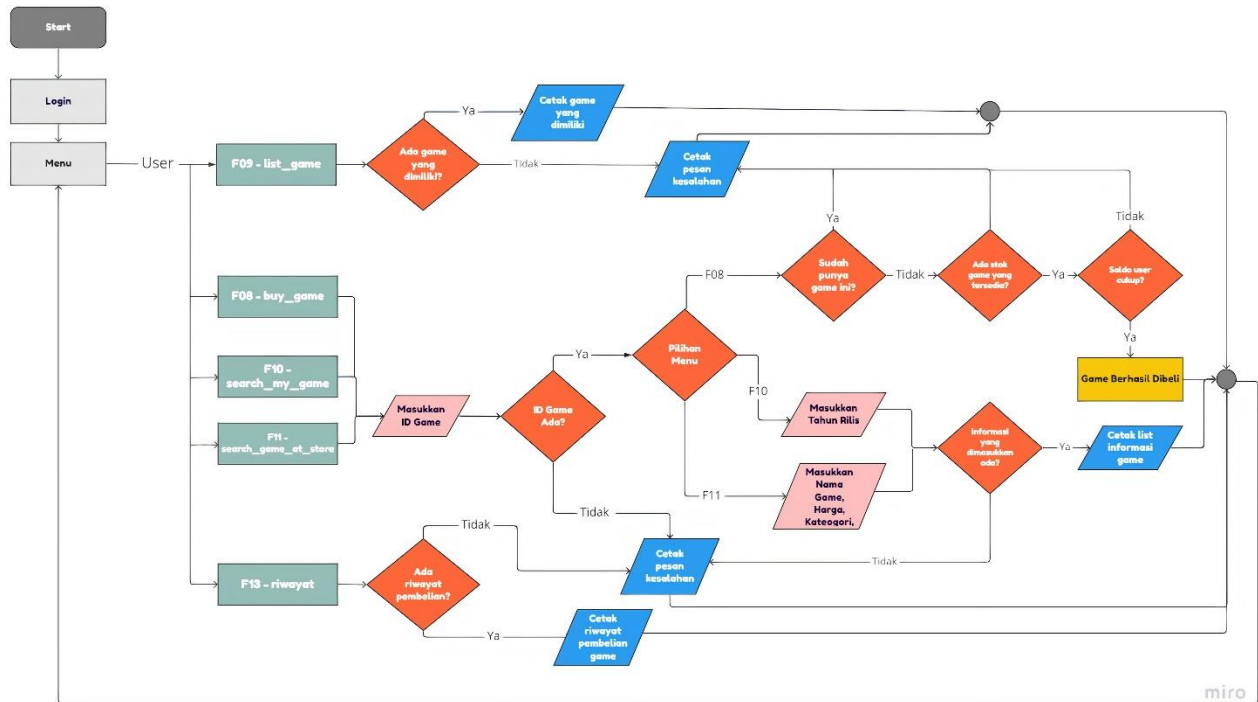
B01	function encrypt (plain : <u>string</u> , a : <u>integer</u> , b : <u>integer</u>) -> <u>string</u> { Mengembalikan hasil enkripsi text plain menggunakan affine cipher dengan key a dan b }
	function decrypt (crypted: <u>string</u> , a : <u>integer</u> , b : <u>integer</u>) -> <u>string</u> { Mengembalikan hasil dekripsi text yang terenkripsi menggunakan affine cipher dengan key a dan b }
	function egcd (a : <u>integer</u> , b : <u>integer</u>) -> (fpb : <u>integer</u> , y : <u>integer</u> , x : <u>integer</u>) { Merupakan fungsi fpb extended yang mengembalikan nilai fpb serta x dan y yang merupakan koefisien identitas Bézout yang memenuhi persamaan $ax + by = fpb(a,b)$ }
	function inverse_modulo (a : <u>integer</u>) -> <u>integer</u> { Mengembalikan hasil inverse modulo yang memenuhi persamaan $x \equiv a^{-1} \pmod{m}$ menggunakan identitas Bézout }
B02	procedure kerangajaib (<u>input</u> tanya : <u>string</u>) { I.S : tanya sembarang F.S : Mencetak jawaban ke layar secara acak }
B03	procedure tictactoe (<u>input</u> X : <u>character</u> , <u>input</u> Y : <u>character</u>) { I.S : X, Y sembarang F.S : Mencetak kondisi akhir game ke layar }
	procedure cetak_papan (<u>input</u> papan : <u>array</u> [0..3] <u>of</u> <u>array</u> [0..3] <u>of</u> <u>character</u>) { I.S : papan terdefinisi F.S : Mencetak status papan ke layar }
	procedure cek_pemain (<u>input</u> i : <u>integer</u>) { I.S : i terdefinisi F.S : Mencetak ke layar apakah sekarang giliran pemain 1 atau 2 }
	function validasi_input (papan : <u>array</u> [0..3] <u>of</u> <u>array</u> [0..3] <u>of</u> <u>character</u> , X : <u>character</u> , Y : <u>character</u>) -> <u>boolean</u> { Mengembalikan nilai true jika X dan Y memenuhi syarat }
	function win_condition (papan : <u>array</u> [0..3] <u>of</u> <u>array</u> [0..3] <u>of</u> <u>character</u>) -> <u>integer</u> { Mengembalikan kode untuk menentukan hasil game; -1 bila game belum berakhir, 0 bila game seri, 1 bila X menang, 2 bila O menang }
	function game_end (kode : <u>integer</u>) -> <u>boolean</u> { Mencetak hasil game dan mengembalikan True apabila game sudah berakhir, False bila game belum berakhir }
	procedure change_admin_or_user (<u>input</u> username : <u>string</u> , <u>input</u> ubah_role : <u>string</u>) { I.S : username dan ubah_role sembarang F.S : Menampilkan hasil perubahan ke layar }

5. DESAIN DEKOMPOSISI ALGORITMIK DAN FUNSIONAL PROGRAM

Gambar 1. Flowchart Program Sebagai Admin + Admin dan User



Gambar 2. Flowchart Program Sebagai User



6. SPESIFIKASI SETIAP MODUL/PROSEDUR/FUNGSI DALAM NOTASI ALGORITMIK

Tabel 11. account.py	
KAMUS	
-	
ALGORITMA PROGRAM UTAMA	
<pre>import data from fungsi_data import *</pre>	
{ REALISASI FUNGSI DAN PROSEDUR }	
F02	<pre>procedure register (<u>input</u> nama : <u>string</u>, <u>input</u> username : <u>string</u>, <u>input</u> password : <u>string</u>) { I.S. nama, username, dan password sembarang F.S. Mencetak pesan ke layar apakah berhasil register atau tidak }</pre> <p>KAMUS LOKAL</p> <pre>nama, username, password : <u>string</u> count, i : <u>integer</u> ada : <u>boolean</u></pre> <p>ALGORITMA</p> <pre><u>output</u>("Masukkan nama: ") <u>input</u>(nama) <u>output</u>("Masukkan username: ") <u>input</u>(username) <u>while</u> (<u>not</u>(validasi_user(username))) <u>do</u> Wait() <u>output</u>("Username tidak valid. Silakan masukan username lagi!") <u>output</u>("Masukkan username: ") <u>input</u>(username) <u>output</u>("Masukkan password: ") <u>input</u>(password) password <- encrypt(password, 5, 9) count <- 0 ada <- <u>false</u> i <u>traversal</u> [1..Length(data.user)] count <- count + 1 <u>if</u> (username = data.user[i][1]) <u>then</u> ada <- <u>true</u> Wait() <u>if</u> (ada) <u>then</u></pre>

	<pre> output("Username ", username, " sudah terpakai, silakan menggunakan username lain.") else if (nama = "") or (username = "") or (password = "") then output("Ada masukan yang kosong. Register gagal!") else output("Username ", username, " telah berhasil register ke dalam 'Binomo'") data.user <- Append(data.user, [count, username, nama, password, "user", 0]) </pre>
	<pre> function validasi_user (username : <u>string</u>) -> <u>boolean</u> { Mengembalikan True jika masukan username benar } </pre> <p>KAMUS LOKAL</p> <pre> i : <u>integer</u> valid : <u>boolean</u> </pre> <p>ALGORITMA</p> <pre> i <- 0 valid <- true while (i < Length(username)) and (valid) do if (ord(username[i].lower()) < 97 or ord(username[i].lower()) > 122) and (username[i] ≠ '-') and (username[i] ≠ '_') and (ord(username[i]) < 48 or ord(username[i]) > 57) then valid <- false i <- i + 1 -> valid </pre>
F03	<pre> function login () -> (<u>boolean</u>, <u>string</u>, <u>string</u>, <u>integer</u>) { Akses masuk ke program jika masukan username dan password benar } </pre> <p>KAMUS LOKAL</p> <pre> login : <u>boolean</u> username, password : <u>string</u> indeks, i, id_user : <u>integer</u> </pre> <p>ALGORITMA</p> <pre> login <- false output("Login: ") output("Masukkan username: ") input(username) output("Masukkan password: ") input(password) indeks <- -9999 i traversal [1..Length(data.user)] if (username = data.user[i][1]) and (password = data.user[i][3]) then indeks <- i </pre>

	<pre> Wait() if (indeks ≠ -9999) then output("Halo ", data.user[indeks][2], "! Selamat datang di 'Binomo'.") login <- true else output("Password atau username salah atau tidak ditemukan. Silahkan login ulang karena akun telah ter-logout!") id_user <- getIndexFromAttribute(username, 1, data.user) id_user <- data.user[id_user][0] -> login, username, password, id_user </pre>
F12	<p>procedure topup (<u>input</u> username : string, <u>input</u> saldo : integer)</p> <p>{ I.S : username dan saldo sembarang F.S : mencetak username dan nilai saldo akhir yang tervalidasi }</p> <p>KAMUS LOKAL</p> <p>v, user_index : <u>integer</u> saldo_akhir, saldo_awal : <u>integer</u></p> <p>ALGORITMA</p> <pre> output("Masukkan username") input(username) user_index <- getIndexFromAttribute (username,1,data.user) try : output("Masukkan saldo") input(saldo) v <- 1 catch ValueError v <- -1 return v if (v = -1) then output("Masukan tidak valid! Saldo harus berupa bilangan bulat.") else saldo_akhir <- 0 Wait() if (Length(saldo) = 0) then output("Masukkan tidak valid") else if (user_index = 0) then output("Username", username, "tidak ditemukan") else saldo_awal <- data.user[user_index][5] </pre>

	<pre> saldo_akhir <- saldo + saldo_awal if (saldo_akhir < 0) then output("Masukkan tidak valid") else if (saldo_awal < saldo_akhir) then output("Top-up berhasil! Saldo",username, "bertambah menjadi", saldo_akhir) else output("Top-up berhasil! Saldo",username, "berkurang menjadi", saldo_akhir) data.user[user_index][5] <- saldo_akhir </pre>
--	--

Tabel 12. showgame.py	
KAMUS	-
ALGORITMA PROGRAM UTAMA	<pre> import data from fungsi_sakti import * </pre>
REALISASI FUNGSI DAN PROSEDUR	

F07	<pre> procedure list_game_toko (<u>input</u> pilih_skema : <u>string</u>) { I.S : pilih_skema terdefinisi F.S : menampilkan list game pada toko berdasarkan tahun+, tahun-, harga+, harga-, atau kosong (tekan tombol Enter)} KAMUS LOKAL pilih_skema : <u>string</u> Temp : <u>array of</u> Game ALGORITMA <u>output</u> ("" Skema sorting: - tahun+ = sorting daftar game secara menaik berdasarkan tahun - tahun- = sorting daftar game secara menurun berdasarkan tahun - harga+ = sorting daftar game secara menaik berdasarkan harga - harga- = sorting daftar game secara menurun berdasarkan harga Jika tidak ingin men-sorting daftar game, silakan tekan Enter. """) <u>output</u>("Masukkan input berupa tahun+, tahun-, harga+, harga-, atau kosong (tekan tombol Enter).") <u>output</u>("Skema sorting: ") <u>input</u>(pilih_skema) lower(pilih_skema) Temp <- [] Wait() <u>depend on</u>(pilih_skema) pilih_skema = "tahun+" : Ascending(3) pilih_skema = "harga+" : Ascending(4) pilih_skema = "tahun-" : Descending(3) pilih_skema = "harga-" : Descending(4) pilih_skema = ""521save16 : Ascending(0) <u>else</u> <u>output</u>("Skema sorting tidak valid!") </pre>
-----	---

	<pre> procedure Ascending (<u>input</u> indeks : <u>integer</u>) { I.S : indeks terdefinisi F.S : mencetak list game di toko dengan terurut membesar } KAMUS LOKAL i, j, IMin : <u>integer</u> Temp : <u>array of</u> Game ALGORITMA <u>output</u>("0. Id Nama Harga Kategori Tahun_rilis Stok") i <u>traversal</u> [2.. Length(data.game)] IMin \leftarrow i j <u>traversal</u> [i+1..Length(data.game)] <u>if</u> (str(data.game[j][indeks]) < str(data.game[IMin][indeks])) <u>then</u> IMin \leftarrow j Temp \leftarrow data.game[i] data.game[i] \leftarrow data.game[IMin] data.game[IMin] \leftarrow Temp CetakSorting() </pre>
	<pre> procedure Descending (<u>input</u> indeks : <u>integer</u>) { I.S : indeks terdefinisi F.S : mencetak list game di toko dengan terurut mengecil } KAMUS LOKAL i, j, IMax : <u>integer</u> Temp : <u>array of</u> Game ALGORITMA <u>output</u>("0. Id Nama Harga Kategori Tahun_rilis Stok") i <u>traversal</u> [2.. Length(data.game)] IMax \leftarrow i j <u>traversal</u> [i+1..Length(data.game)] <u>if</u> (str(data.game[j][indeks]) > str(data.game[IMax][indeks])) <u>then</u> IMax \leftarrow j Temp \leftarrow data.game[i] data.game[i] \leftarrow data.game[IMax] data.game[IMax] \leftarrow Temp CetakSorting() </pre>
	<pre> procedure CetakSorting (<u>output</u> count : <u>integer</u>) { I.S : data.game berisi data csv F.S : mencetak data.game ke layar secara terurut } KAMUS LOKAL count, i, j : <u>integer</u> ALGORITMA </pre>

	<pre> count <- 0 i <u>traversal</u> [2..Length(data.game)] count <- count + 1 <u>output</u>(count, ".", end=" ") j <u>traversal</u> [1..6] <u>if</u> (j ≠ 5) <u>then</u> <u>depend on</u> (j) j = 0 <u>or</u> j = 1 : <u>output</u>(data.game[i][j], end=" ") j = 2 : <u>output</u>(data.game[i][4], end=" ") <u>else</u> <u>output</u>(data.game[i][j-1], end=" ") <u>else</u> <u>output</u>(data.game[i][j]) </pre>
F09	<p>procedure list_game (<u>input</u> user_id : <u>string</u>) { I.S : user_id sembarang F.S : Mencetak semua game yang dimiliki oleh user dengan user_id tersebut ke layar }</p> <p>KAMUS LOKAL game_yang_dimiliki : <u>array of string</u> i : <u>integer</u></p> <p>ALGORITMA game_yang_dimiliki = [] i <u>traversal</u> [1..Length(data.kepemilikan)] <u>if</u> (user_id = data.kepemilikan[i][1]) <u>then</u> game_yang_dimiliki = Append(game_yang_dimiliki, data.kepemilikan[i][1])</p> <p><u>if</u> (Length(game_yang_dimiliki) = 0) <u>then</u> <u>output</u>("Anda belum memiliki game! Silahkan beli dengan command F08!") <u>else</u> count <- 1 Wait() <u>output</u>("Daftar Game:") <u>output</u>("0. Id Nama Kategori Tahun_rilis Harga Stok") i <u>traversal</u> [1..Length(game_yang_dimiliki)] game_index <- getIndexFromAttribute(game_yang_dimiliki[i], 0, data,game) <u>output</u>(count, ". ", data.game[game_index][0], " ", data.game[game_index][1], " ", data.game[game_index][2], " ", data.game[game_index][3], " ", data.game[game_index][4]) count <- count + 1</p>

F10	<pre> procedure search_my_game (<u>input</u> id_user : <u>string</u>, <u>input</u> ID : <u>string</u>, <u>input</u> tahun_rilis : <u>string</u>) { I.S. id_user terdefinisi, ID dan tahun_rilis sembarang F.S. Menampilkan hasil pencarian game ke layar berdasarkan ID dan tahun rilis } KAMUS LOKAL ID, tahun_rilis : <u>string</u> count, i, j, k : <u>integer</u> my_game : <u>array of string</u> ALGORITMA <u>output</u>("Masukkan ID Game: ") <u>input</u>(ID) <u>output</u>("Masukkan Tahun Rilis Game: ") <u>input</u>(tahun_rilis) <u>output</u>("Daftar game pada inventory yang memenuhi kriteria:") Wait() count <- 1 my_game <- [] i <u>traversal</u> [1..Length(data.kepemilikan)] <u>if</u> (str(id_user) = str(data.kepemilikan[i][1])) <u>and</u> (ID = str(data.kepemilikan[i][0]) <u>or</u> ID = "") <u>then</u> my_game <- Append(my_game, str(data.kepemilikan[i][0])) j <u>traversal</u> [1..Length(my_game)] k <u>traversal</u> [1..Length(data.game)] <u>if</u> (my_game[j] = data.game[k][0]) <u>and</u> ((tahun_rilis = data.game[k][3]) <u>or</u> (tahun_rilis = "")) <u>then</u> <u>if</u> (count = 1) <u>then</u> <u>output</u>("0. Id Nama Kategori Tahun_rilis Harga Stok") CetakGame(k, count) count <- count + 1 <u>if</u> (count = 1) <u>then</u> <u>output</u>("Tidak ada game pada inventory-mu yang memenuhi kriteria") procedure CetakGame (<u>input</u> indeks : <u>integer</u>, <u>input</u> count : <u>integer</u>) { I.S. indeks dan count terdefinisi, F.S. Mencetak hasil pencarian ke layar } KAMUS LOKAL j : <u>integer</u> </pre>
-----	---

	<p>ALGORITMA</p> <pre> output(count, ".", end = " ") j traversal [1..6] if (j ≠ 5) then output(data.game[indeks][j], end=" ") else output(data.game[indeks][j]) </pre>
F11	<p>procedure search_game_at_store (<u>input</u> ID : <u>string</u>, <u>input</u> nama : <u>string</u>, <u>input</u> harga : <u>string</u>, <u>input</u> kategori : <u>string</u>, <u>input</u> tahun_rilis : <u>string</u>) { I.S. ID, nama, harga, kategori, dan tahun_rilis sembarang F.S. Menampilkan hasil pencarian game ke layar berdasarkan ID, nama, harga, kategori, dan tahun_rilis }</p> <p>KAMUS LOKAL</p> <p>ID, nama, harga, kategori, tahun_rilis : <u>string</u> count, i : <u>integer</u></p> <p>ALGORITMA</p> <pre> output("Masukkan ID Game: ") input(ID) output("Masukkan Nama Game: ") input(nama) output("Masukkan Harga Game: ") input(harga) output("Masukkan Kategori Game: ") input(kategori) output("Masukkan Tahun Rilis Game: ") input(tahun_rilis) output("Daftar game pada inventory yang memenuhi kriteria:") Wait() count <- 1 i traversal [1..Length(data.game)] if (i = 0) then output("0. Id Nama Kategori Tahun_rilis Harga Stok") { masukan masing-masing masukan parameter boleh kosong } if (ID = data.game[i][0] or ID = '') and (nama = data.game[i][1] or nama = '') and (kategori = data.game[i][2] or kategori = '') and (tahun_rilis = data.game[i][3] or tahun_rilis = '') and (harga = data.game[i][4] or harga = '') then CetakGame(1, count) count <- count + 1 if (count = 1) then output("Tidak ada game pada inventory-mu yang </pre>

	memenuhi kriteria")
F13	<pre> procedure riwayat (<u>input</u> user_id : <u>string</u>) { I.S : user_id terdefinisi F.S : menampilkan riwayat pembelian game dari user_id Terkait } KAMUS LOKAL riwayat_index : <u>integer</u> riwayat_user : <u>array of</u> Riwayat ALGORITMA riwayat_user <- [] i <u>traversal</u> [1..Length(data.riwayat)] <u>if</u> (str(user_id) = str(i[3])) <u>then</u> riwayat_user[i] <- Append(riwayat_user, data.riwayat[i]) <u>if</u> (riwayat_user = []) <u>then</u> <u>output</u>("Maaf, kamu tidak memiliki riwayat pembelian game. Ketik perintah F08 untuk membeli.") <u>else</u> <u>output</u>("0. ID Nama Game Harga Tahun_Beli") i <u>traversal</u> [1..Length(riwayat_user)] <u>if</u> (str(user_id) == str(riwayat_user[i][3])) <u>then</u> <u>output</u>(i, ". ", riwayat_user[i][0], " ", riwayat_user[i][1], " ", riwayat_user[i][2], " ", riwayat_user[i][4]) </pre>

Tabel 13. modifygame.py	
KAMUS	
-	
ALGORITMA PROGRAM UTAMA	
<pre> <u>import</u> data <u>from</u> fungsi_sakti <u>import</u> * </pre>	
REALISASI FUNGSI DAN PROSEDUR	
F04	<pre> procedure tambah_game (<u>input</u> nama, kategori, tahun_rilis, harga, stok : <u>string</u>) { I.S : nama, kategori, tahun_rilis, harga, stok sembarang F.S : seluruh data atribut game terdefinisi } KAMUS LOKAL banyak_data : <u>integer</u> id_game : <u>string</u> ALGORITMA banyak_data <- Length(data.game) </pre>

	<pre> <u>depend on</u> (Length(banyak_data)) Length(banyak_data) = 1 : id_game <- "G00" + banyak_data Length(banyak_data) = 2 : id_game = "G0" + banyak_data else id_game <- "G" + banyak_data <u>iterate</u> <u>output</u> ("Masukkan nama game: ") <u>input</u> (nama) <u>output</u> ("Masukkan kategori: ") <u>input</u> (kategori) <u>output</u> ("Masukkan tahun rilis: ") <u>input</u> (tahun_rilis) <u>output</u> ("Masukkan harga: ") <u>input</u> (harga) <u>output</u> ("Masukkan nama stok: ") <u>input</u> (stok) <u>stop</u> ((Length(nama) ≠ 0) and (Length(kategori) ≠ 0) and (Length(tahun_rilis) ≠ 0) and (Length(harga) ≠ 0) and (Length(stok) ≠ 0)) Clear() <u>output</u> ("Mohon masukkan semua informasi mengenai game agar dapat disimpan BNMO.") data.game <- Append(data.game, [id_game, nama, kategori, tahun_rilis, harga, stok]) <u>output</u> ("Selamat! Berhasil menambahkan game", nama) </pre>
F05	<p>procedure ubah_game (<u>input</u> game_id, nama, kategori, tahun_rilis, harga : <u>string</u>)</p> <p>{ I.S : game_id, nama, kategori, tahun_rilis, harga sembarang F.S : menampilkan game_id, nama, kategori, tahun_rilis, dan harga yang telah diubah jika masukkan tidak kosong }</p> <p>KAMUS LOKAL</p> <p>game_id = <u>string</u> id_index : <u>integer</u></p> <p>ALGORITMA</p> <pre> <u>output</u> ("Masukkan ID Game") <u>input</u> (game_id) id_index <- getIndexFromAttribute (game_id, 0, data.game) <u>if</u> (id_index = 0) <u>then</u> <u>output</u> ("Tidak ada game dengan ID tersebut!") else <u>output</u> ("") <u>output</u> ("Data sebelum diubah adalah: ") <u>output</u> ("[Nama Game, Kategori, Tahun Rilis, Harga, Stok]") i <u>traversal</u> [0..5] <u>depend on</u> (i) </pre>

	<pre> i = 0 : <u>output</u> ("[" , end="") i = 5 : <u>output</u> (str(data.game[id_index][i]), "]") else <u>output</u> (str(data.game[id_index][i]), ", ") <u>output</u>("") <u>output</u> ("Masukkan nama game: ") <u>input</u> (nama) <u>output</u> ("Masukkan kategori: ") <u>input</u> (kategori) <u>output</u> ("Masukkan tahun rilis: ") <u>input</u> (tahun_rilis) <u>output</u> ("Masukkan harga: ") <u>input</u> (harga) <u>output</u>("") if (int(tahun_rilis) ≥ 0 <u>and</u> int(harga) ≥ 0) <u>then</u> if (Length(nama) ≠ 0) <u>then</u> data.game[id_index][1] <- nama if (Length(kategori) ≠ 0) <u>then</u> data.game[id_index][2] <- kategori if (Length(tahun_rilis) ≠ 0) <u>then</u> data.game[id_index][3] <- int(tahun_rilis) if (Length(harga) ≠ 0) <u>then</u> data.game[id_index][4] <- int(harga) Wait() <u>output</u>("") <u>output</u> ("Data setelah diubah adalah: ") <u>output</u> ("[Nama Game, Kategori, Tahun Rilis, Harga, Stok]") i <u>traversal</u> [0..5] <u>depend on</u> (i) i = 0 : <u>output</u> ("[" i = 5 : <u>output</u> (str(data.game[id_index][i]), "]") else <u>output</u> (str(data.game[id_index][i]), ", ") else <u>output</u> ("Masukkan salah! Silahkan ulangi.") </pre>
F06	<p>procedure ubah_stok (<u>input</u> game_id : <u>string</u> <u>input</u> jumlah_tambahan : <u>integer</u>) { I.S : game_id dan jumlah_tambahan sembarang F.S : mengupdate stok game dengan game_id input sebanyak tambahan sesuai input jumlah_tambahan kemudian dicetak perubahan / tidak perubahannya }</p> <p>KAMUS LOKAL game_id, game_nama : <u>string</u> game_index, game_stok, game_stok_baru : <u>integer</u></p> <p>ALGORITMA</p>

	<pre> output("Masukkan ID game: ") input(game_id) game_index = getIndexFromAttribute(game_id, 0, data.game) if (game_index ≠ 0) then jumlah_tambahan <- 0 while (jumlah_tambahan = 0) do try output("Masukkan jumlah: ") input(jumlah_tambahan) catch ValueError output("Mohon masukkan bilangan bulat!") game_nama <- data.game[game_index][1] game_stok <- int(data.game[game_index][5]) game_stok_baru <- game_stok + jumlah_tambahan Wait() if (game_stok_baru ≥ 0) then data.game[game_index][5] <- str(game_stok_baru) if (game_stok_baru > game_stok) then output("Stok game ", game_nama, " berhasil ditambahkan. Stok sekarang: ", game_stok_baru) else: output("Stok game ", game_nama, " berhasil dikurangi. Stok sekarang: ", game_stok_baru) else: output("Stok game ", game_nama, " gagal dikurangi karena stok kurang. Stok sekarang: ", game_stok, " (< ", abs(jumlah_tambahan), ")") else: output("Tidak ada game dengan ID tersebut!") </pre>
F08	<p>procedure buy_game (<u>input</u> user_id, game_id : <u>string</u>) { I.S : game_id dan user_id sembarang F.S : menambahkan game dengan game_id sesuai input ke dalam kepemilikan user dengan user_id sesuai input, mengurangi saldo user, dan mengurangi stok game bila berhasil terbeli }</p> <p>KAMUS LOKAL game_id, game_nama, user_id : <u>string</u> game_index, user_index, game_harga, game_stok, user_saldo : <u>integer</u> punyaGame : <u>boolean</u></p> <p>ALGORITMA output("Masukkan ID game: ") input(game_id) game_index <- getIndexFromAttribute(game_id, 0, data.game)</p>

	<pre> user_index <- getIndexFromAttribute(user_id, 0, data.user) if(game_index ≠ 0) then game_nama <- data.game[game_index][1] game_harga <- int(data.game[game_index][4]) game_stok <- int(data.game[game_index][5]) user_id <- data.user[user_index][0] user_saldo <- int(data.user[user_index][5]) punyaGame <- False Wait() i traversal [1..Length(data.kepemilikan)] if (data.kepemilikan[i][0] == game_id and data.kepemilikan[i][1] == user_id) then punyaGame = True if (not(punyaGame)) then if(game_stok > 0) then if(user_saldo ≥ game_harga) then user_saldo <- user_saldo - game_harga game_stok <- game_stok - 1 data.user[user_index][5] <- str(user_saldo) data.kepemilikan <- Append(data.kepemilikan, [game_id, user_id]) data.game[game_index][5] <- str(game_stok) data.riwayat <- Append(data.riwayat, [game_id, game_nama, str(game_harga), user_id, '2022']) output("Game ", game_nama, " berhasil dibeli!") else output("Saldo anda tidak cukup untuk membeli Game tersebut!") else output("Stok Game tersebut sedang habis!") else output("Anda sudah memiliki Game tersebut!") else output("Game tidak ditemukan!") </pre>
--	--

Tabel 14. system.py	
KAMUS	-
ALGORITMA PROGRAM UTAMA	<pre> import data, os, sys, argparse from fungsi_sakti import * </pre>

REALISASI FUNGSI DAN PROSEDUR

F15	<pre>procedure load(<u>input</u> nama_folder : <u>string</u>) { I.S : Nama folder penyimpanan sembarang F.S : Membaca seluruh file csv pada folder dan menyimpannya dalam array } KAMUS LOKAL dirpath, firnames, filenames, folders : Array[] of <u>string</u> dir_path, args.nama_folder : <u>string</u> Found : <u>boolean</u> ALGORTIMA parser <- argparse.ArgumentParser() parser.add_argument('nama_folder', help="nama folder dimana file csv berada", nargs='?', const='') args <- parser.parse_args() <u>if</u> (args.nama_folder) <u>then</u> dir_path <- os.path.dirname(os.path.realpath(__file__)) (dirpath, dirnames, filenames) <u>traversal</u> [os.walk(dir_path)] folders <- dirnames <u>break</u> Found <- <u>false</u> <u>i traversal</u> [1..Length(folders)] <u>if</u> (args.nama_folder = folders[i]) <u>then</u> Found <- <u>true</u> <u>if</u> Found <u>then</u> data.readAllCSV(args.nama_folder) <u>else</u> <u>output</u>("Folder ", args.nama_folder , " tidak ditemukan.") sys.exit() <u>else</u> <u>output</u>("Tidak ada nama folder yang diberikan!\nUsage: python program_binomo.py <nama_folder>") sys.exit()</pre>
F16	<pre>procedure save(<u>input</u> loc : <u>string</u>) { I.S Lokasi sembarang, folder belum tentu ada pada lokasi tersebut F.S Folder terbuat sesuai lokasi dari input user dan didalam folder } KAMUS LOKAL loc : <u>string</u> dirpath, firnames, filenames, folders : Array[] of <u>string</u> dir_path, args.nama_folder : <u>string</u> ALGORITMA <u>output</u>("Masukkan nama folder penyimpanan: ")</pre>

	<pre> <u>input</u>(loc) dir_path <- os.path.dirname(os.path.realpath(__file__)) (dirpath, dirnames, filenames) <u>traversal</u> [os.walk(dir_path)] <u>then</u> folders <- dirnames <u>break</u> path <- os.path.join(dir_path, loc) <u>if</u>(loc in folders)<u>then</u> <u>pass</u> <u>else</u> Clear() os.makedirs(path) writeToCSV(path, data.user, "user") writeToCSV(path, data.game, "game") writeToCSV(path, data.kepemilikan, "kepemilikan") writeToCSV(path, data.riwayat, "riwayat") <u>output</u>("File berhasil di-save di folder ", loc) </pre> <p>procedure writeToCSV(<u>input</u> path : <u>string</u> <u>input</u> data_array : <u>Array</u> of <u>Array</u> of <u>string</u> <u>input</u> data_name : <u>string</u>) { I.S : Path, data_array, dan data_name sembarang F.S : Terbentuk file <data_name>.csv yang berisikan data dalam array data_array } KAMUS LOKAL baris, baris_csv : <u>string</u> f : SEQFILE of <u>string</u></p> <p>ALGORITMA file_name <- data_name + ".csv"</p> <pre> <u>assign</u>(f, file_name) <u>rewrite</u>(f) i <u>traversal</u> [1..Length(data_array)] baris_csv <- "" j <u>traversal</u> [1..Length(data_array[i])] baris_csv <- baris_csv + data_array[i][j] + ';' baris_csv <- baris_csv + '\n' <u>write</u>(f, baris_csv) <u>close</u>(f) </pre>
F17	<p>procedure Exit (<u>input</u> pilihan : <u>string</u>) { I.S : Pilihan terdefinisi F.S : Mengeksekusi pilihan dan keluar dari aplikasi }</p> <p>KAMUS LOKAL pilihan : <u>string</u> valid : <u>boolean</u></p>

	ALGORITMA <pre> valid <- <u>false</u> while not <u>valid</u> <u>do</u> <u>output</u>("Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n): ") <u>input</u>(pilihan) <u>lower</u>(pilihan)l <u>depend on</u>(pilihan) pilihan = 'y' : <u>save</u>(), <u>valid</u> <- <u>true</u>) pilihan = 'n' : <u>valid</u> <- <u>true</u> <u>else</u> Ckear() <u>output</u>("Masukan salah!") <u>output</u>("Dadah!") <u>sys.exit</u>() </pre>
--	--

Tabel 15. help.py	
KAMUS	-
ALGORITMA PROGRAM UTAMA	<pre> <u>from</u> <u>fungsi_sakti</u> <u>import</u> * <u>import</u> <u>data</u> </pre>
{ REALISASI FUNGSI/PROSEDUR }	
procedure Help(<u>input</u> admin : <u>boolean</u>)	
{ I.S : Admin terdefinisi sembarang	
F.S : Menampilkan panduan penggunaan aplikasi }	
KAMUS LOKAL	
ALGORITMA	
Wait()	
<u>if</u> (admin)then	
<u>output</u> (""	
===== HELP =====	
F02 - register - Untuk melakukan registrasi user baru	
F03 - login - Untuk melakukan login ke dalam sistem	
F04 - tambah_game - Untuk menambah game yang dijual pada toko	
F05 - ubah_game - Untuk mengubah game pada toko game	
F06 - ubah_stok - Mengubah stok sebuah game pada toko dilakukan melalui input ID dan besar perubahan stok yang ingin dilakukan.	
F07 - list_game_toko - Untuk melihat list game yang dijual pada toko	
F11 - search_game_at_store - Untuk mencari game di toko berdasarkan ID, Nama Game, Harga, Kategori dan Tahun Rilis	
F12 - topup - untuk menambahkan saldo kepada User	
F16 - save - untuk melakukan penyimpanan data ke dalam file setelah dilakukan perubahan	
F17 - exit - untuk keluar dari aplikasi""")	

```

else
    output ("""
===== HELP =====
F02 - login - Untuk melakukan login ke dalam sistem
F07 - list_game_toko - Untuk melihat list game yang dijual pada toko
F08 - buy_game - untuk membeli game yang diinginkan pengguna
F09 - list_game - memberikan daftar game yang dimiliki pengguna
F10 - search_my_game - untuk mendapatkan informasi game sesuai dengan
                        query yang diminta oleh pengguna pada
                        inventory
F11 - search_game_at_store - Untuk mencari game di toko berdasarkan
                            ID, Nama Game, Harga, Kategori dan Tahun
                            Rilis
F13 - riwayat - untuk melihat riwayat pembelian game yang dilakukan
                        pengguna
F16 - save - untuk melakukan penyimpanan data ke dalam file setelah
                        dilakukan perubahan
F17 - exit - untuk keluar dari aplikasi""")

```

Tabel 16. misc.py

KAMUS

ALGORITMA PROGRAM UTAMA

```

import time, data
from fungsi_sakti import *

```

REALISASI FUNGSI DAN PROSEDUR

B01	<pre> function encrypt (plain : string, a : <u>integer</u>, b : <u>integer</u>) -> string { Mengembalikan hasil enkripsi text plain menggunakan affine cipher dengan key a dan b } KAMUS LOKAL crypted, plain : string chara : character a, b, i : integer ALGORITMA crypted <- "" i <u>traversal</u> [1..Length(plain)] chara = plain[i] <u>if</u> (ord(chara) ≥ 48 <u>and</u> ord(chara) ≤ 57) <u>or</u> (ord(chara) = 32) <u>then</u> { Character angka atau spasi diabaikan } crypted <- crypted + chara <u>if</u> (ord(chara) ≥ 97 <u>and</u> ord(chara) ≤ 122) <u>then</u> crypted <- crypted + chr((a*(ord(chara) - ord('a'))+b) <u>mod</u> 26 + ord('a')) </pre>
-----	---

	<pre> if (ord(chara) ≥ 65 and or(chara) ≤ 90) then crypted <- crypted + chr((a*(ord(chara) - ord('A'))+b)mod26 + ord('A')) -> crypted </pre>
	<pre> function decrypt (crypted: <u>string</u>, a : <u>integer</u>, b : <u>integer</u>) -> string { Mengembalikan hasil dekripsi text yang terenkripsi menggunakan affine cipher dengan key a dan b } </pre> <p>KAMUS LOKAL</p> <pre> crypted, plain : string chara : character a, b, i : integer </pre> <p>ALGORITMA</p> <pre> plain <- "" i <u>traversal</u> [1..Length(crypted)] chara = crypted[i] if (ord(chara) ≥ 48 and ord(chara) ≤ 57) or (ord(chara) = 32) then plain <- plain + chara if (ord(chara) ≥ 97 and or(chara) ≤ 122) then plain <- plain + chr(((inverse_modulo(a)*(ord(chara) - ord('a') - b)) mod 26) + ord('a')) if (ord(chara) ≥ 65 and or(chara) ≤ 90) then plain <- plain + chr(((inverse_modulo(a)*(ord(chara) - ord('A') - b)) mod 26) + ord('A')) -> plain </pre> <pre> function egcd (a : <u>integer</u>, b : <u>integer</u>) -> (fpb : <u>integer</u>, y : <u>integer</u>, x : <u>integer</u>) { Merupakan fungsi fpb extended yang mengembalikan nilai fpb serta x dan y yang merupakan koefisien identitas Bézout yang memenuhi persamaan ax + by = fpb(a,b) } </pre> <p>KAMUS LOKAL</p> <pre> type hasil : < fpb : <u>integer</u>; y : <u>integer</u>; x : <u>integer</u> > a, b : integer </pre> <p>ALGORITMA</p> <pre> if a = 0 then hasil.fpb <- b hasil.y <- 0 hasil.x <- 1 -> hasil else hasil <- egcd(b mod a, a) -> hasil </pre>

	<p>function <u>inverse_modulo</u> (<u>a</u> : <u>integer</u>) -> <u>integer</u> { Mengembalikan hasil inverse modulo yang memenuhi persamaan $x \equiv a^{-1} \pmod{m}$ menggunakan identitas Bézout }</p> <p>KAMUS LOKAL type hasil : < fpb : <u>integer</u>; y : <u>integer</u>; x : <u>integer</u> > a, m : integer</p> <p>ALGORITMA m <- 26 hasil <- egcd(a, m) <u>if</u> hasil.fpb \neq 1 <u>then</u> -> None <u>else</u> -> x <u>mod</u> m</p>
B02	<p>procedure kerangajaib (<u>input</u> tanya : <u>string</u>) { I.S : Tanya sembarang F.S : Mencetak jawaban ke layar secara acak }</p> <p>KAMUS LOKAL tanya : <u>string</u> jawaban : <u>array</u> [1..6] <u>of</u> <u>string</u> LCG, i : <u>integer</u></p> <p>ALGORITMA <u>output</u>("Apa pertanyaannmu? ") <u>input</u>(tanya) jawaban <- ["Ya", "Tidak", "Bisa Jadi", "Mungkin", "Tentunya", "Tidak mungkin"] LCG <- round((time.time()*Length(tanya) + 2022) <u>mod</u> (Length(jawaban)-1)) { Menggunakan algoritma LCG yaitu $(a*x + c) \% m$. Pada program ini, a = waktu saat ini (dalam satuan detik), x = jumlah karakter pada variabel tanya, c = tahun pembuatan program, yaitu 2022, dan m = jumlah kata pada array jawaban dikurang 1 (karena indeks dimulai dari 0) }</p> <p> i <u>traversal</u> [1..Length(jawaban)] <u>if</u> (LCG = i) <u>then</u> <u>output</u>("Menunggu jawaban dalam beberapa detik...") Wait() <u>output</u>(jawaban[i])</p>
B03	<p>procedure tictactoe (<u>input</u> X : <u>character</u>, <u>input</u> Y : <u>character</u>) { I.S : X, Y sembarang F.S : Mencetak kondisi akhir game ke layar }</p> <p>KAMUS LOKAL papan : <u>array</u> [0..3] <u>of</u> <u>array</u> [0..3] <u>of</u> <u>character</u></p>

	<pre> i : integer X, Y : character selesai : boolean ALGORITMA output("""" Legenda: # Kosong X Pemain 1 O Pemain 2 """) cetak_papan(papan) i <- 0 selesai <- false while (i < 9) and (not(selesai)) do output("X: ") input(X) output("Y: ") input(Y) while (not(validasi_input(papan, X, Y))) do Clear() cetak_papan(papan) output("Kotak tidak valid.") cek_pemain(i) output("X: ") input(X) output("Y: ") input(Y) if (i mod 2) then { Pemain 2 ('O') } papan[int(X)][int(Y)] <- 'O' else { Pemain 1 ('X') } papan[int(X)][int(Y)] <- 'X' Clear() cetak_papan(papan) if (game_end(win_condition(papan))) then Wait() selesai <- true i <- i + 1 </pre>
	<pre> procedure cetak_papan (input papan : array [0..3] of array [0..3] of character) { I.S : papan terdefinisi F.S : Mencetak status papan ke layar } </pre> <p>KAMUS LOKAL</p> <pre> row, col : integer </pre>

	<p>ALGORITMA</p> <pre> output("Status Papan") row <u>traversal</u> [1..3] col <u>traversal</u> [1..3] <u>output</u>(papan[row][col], <u>end</u>="") <u>output</u>("") </pre>
	<p>procedure cek_pemain (<u>input</u> i : <u>integer</u>) { I.S : i terdefinisi F.S : Mencetak ke layar apakah sekarang giliran pemain 1 atau 2 } KAMUS LOKAL ALGORITMA <pre> <u>if</u> (i <u>mod</u> 2) <u>then</u> <u>output</u>("Giliran Pemain 2 ('O') ") <u>else</u> <u>output</u>("Giliran Pemain 1 ('X') ") </pre> </p>
	<p>function <u>validasi_input</u> (papan : <u>array</u> [0..3] <u>of</u> <u>array</u> [0..3] <u>of</u> <u>character</u>, X : <u>character</u>, Y : <u>character</u>) -> <u>boolean</u> { Mengembalikan nilai true jika X dan Y memenuhi syarat } KAMUS LOKAL ALGORITMA <pre> <u>try</u> X <- <u>int</u>(X) Y <- <u>int</u>(Y) <u>catch</u> <u>ValueError</u> -> <u>false</u> <u>if</u> ((X < 1) <u>or</u> (X > 3)) <u>or</u> ((Y < 1) <u>or</u> (Y > 3)) <u>or</u> (papan[X][Y] = 'X') <u>or</u> (papan[X][Y] = 'O') <u>then</u> -> <u>false</u> <u>else</u> -> <u>true</u> </pre> </p>
	<p>function <u>win_condition</u> (papan : <u>array</u> [0..3] <u>of</u> <u>array</u> [0..3] <u>of</u> <u>character</u>) -> <u>integer</u> { Mengembalikan kode untuk menentukan hasil game; -1 bila game belum berakhir, 0 bila game seri, 1 bila X menang, 2 bila O menang} KAMUS LOKAL <pre> seri : <u>boolean</u> i, j : <u>integer</u> </pre> ALGORITMA <pre> seri <- <u>true</u> </pre> </p>

```

i traversal [1..3]
  j traversal [1..3]
    if papan[i][j] = '#' then
      seri <- false
if((papan[1][1] = 'X' and papan[2][2] = 'X' and papan[3][3] =
  'X') or (papan[1][3] = 'X' and papan[2][2] = 'X' and
  papan[3][1] = 'X')) then
  -> 1

if((papan[1][1] = 'O' and papan[2][2] = 'O' and papan[3][3] =
  'O') or (papan[1][3] = 'O' and papan[2][2] = 'O' and
  papan[3][1] == 'O')) then
  -> 2

Length(papan)
  if(papan[i][1] = 'X' and papan[i][2] = 'X' and papan[i][3]
    = 'X') then
    -> 1
  else if(papan[i][1] = 'O' and papan[i][2] = 'O' and
    papan[i][3] = 'O') then
    -> 2

i traversal [1..3]
  if(papan[1][i] = 'X' and papan[2][i] = 'X' and papan[3][i]
    = 'X') then
    -> 1
  else if(papan[1][i] = 'O' and papan[2][i] = 'O' and
    papan[3][i] = 'O') then
    -> 2

if seri then
  -> 0
-> -1

```

```

function game_end (x : integer) -> boolean
{ Mencetak hasil game dan mengembalikan True apabila game sudah
berakhir, False bila game belum berakhir }

```

KAMUS LOKAL

ALGORITMA

```

if(x = 0) then
  output ("\nGame berakhir seri!")
else if(x = 2) then
  output ("\nGame dimenangkan oleh player 'O'")
else if(x = 1) then
  output ("\nGame dimenangkan oleh player 'X'")

if(x >= 0) then
  -> true

```

	<pre> else -> false </pre>
	<pre> procedure change_admin_or_user (<u>input</u> username : <u>string</u>, <u>input</u> ubah_role : <u>string</u>) { I.S : username dan ubah_role sembarang F.S : Menampilkan hasil perubahan ke layar } KAMUS LOKAL username, ubah_role : <u>string</u> valid : <u>boolean</u> i : <u>integer</u> ALGORITMA <u>output</u>("Masukkan username yang ingin diganti rolenya: ") <u>input</u>(username) <u>output</u>("Role ingin diganti menjadi admin atau user?: ") <u>input</u>(ubah_role) Wait() valid <- <u>false</u> i <u>traversal</u> [1..Length(data.user)] <u>if</u> (data.user[i][1] = username) <u>then</u> data.user[i][4] <- ubah_role <u>output</u>("Berhasil diubah!") valid <- <u>true</u> <u>if</u> (<u>not</u>(valid)) <u>then</u> <u>output</u>("Username tidak ditemukan.") </pre>

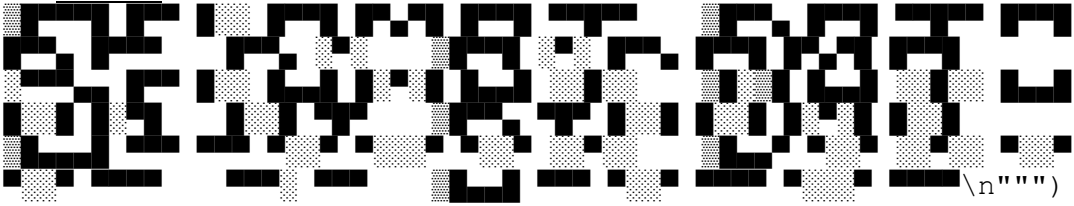
Tabel 16. menu.py	
KAMUS	<pre> cek_login : <u>boolean</u> username, password, id_user : <u>string</u> </pre>
ALGORITMA PROGRAM UTAMA	<pre> <u>import</u> os, data <u>from</u> fungsi_sakti <u>import</u> * <u>from</u> account <u>import</u> * <u>from</u> help <u>import</u> * <u>from</u> misc <u>import</u> * <u>from</u> modifygame <u>import</u> * <u>from</u> showgame <u>import</u> * <u>from</u> system <u>import</u> * cek_login <- <u>false</u> username <- "" password <- "" id_user <- "" </pre>

REALISASI FUNGSI DAN PROSEDUR

1. **procedure** menu_awal ()
{ I.S : Menampilkan selamat datang dan memanggil fungsi login
F.S : Menampilkan menu baru jika berhasil login }

KAMUS LOKAL
username, password, id_user : string
cek_login : boolean

ALGORITMA
global cek_login, username, password, id_user

output(""

\\n"")

cek_login, username, password, id_user <- login()

if (cek_login) then
Clear()
menu()
else
if (username = "") and (password = "") then
sys.exit()
else
menu_awal()
2. **procedure** menu (input pilihan : string)
{ I.S : Pilihan sembarang dan menampilkan menu pilihan ke layar
F.S : Memanggil fungsi sesuai masukan }

KAMUS LOKAL
cek_login, admin : boolean
pilihan, username, password, id_user : string
hanya_admin, hanya_user : string

ALGORITMA
global cek_login, username, password, id_user
output(""\\nDaftar menu:

F02 - register
F03 - login
F04 - tambah_game
F05 - ubah_game
F06 - ubah_stok
F07 - list_game_toko

```

F08 - buy_game
F09 - list_game
F10 - search_my_game
F11 - search_game_at_store
F12 - topup
F13 - riwayat
F14 - Help
F16 - save
F17 - exit
B02 - Magic Conch Shell
B03 - Game Tic-Tac-Toe
T01 - change_admin_or_user
""")

admin <- cek_role(username)
hanya_admin <- "Maaf, anda tidak memiliki izin untuk
                menjalankan perintah berikut. Mintalah ke
                administrator untuk melakukan hal tersebut."
hanya_user <- "Maaf, anda harus menjadi user untuk melakukan
               hal tersebut."

output("Masukkan pilihan menu (F02 - F17/B02 - B03/T01): ")
input(pilihan)
upper(pilihan)
Clear()

depend_on (pilihan)
  pilihan = "F02" : if (admin) then register() else
                    output(hanya_admin)
  pilihan = "F03" : cek_login, username, password, id_user
                    <- login()
  pilihan = "F04" : if (admin) then tambah_game() else
                    output(hanya_admin)
  pilihan = "F05" : if (admin) then ubah_game() else
                    output(hanya_admin)
  pilihan = "F06" : if (admin) then ubah_stok() else
                    output(hanya_admin)
  pilihan = "F07" : list_game_toko()
  pilihan = "F08" : if (admin) then output(hanya_user) else
                    buy_game(id_user)
  pilihan = "F09" : if (admin) then output(hanya_user) else
                    list_game(id_user)
  pilihan = "F10" : if (admin) then output(hanya_user) else
                    search_my_game()
  pilihan = "F11" : search_game_at_store()
  pilihan = "F12" : if (admin) then admin() else
                    output(hanya_admin)
  pilihan = "F13" : if (admin) then output(hanya_user) else
                    riwayat(id_user)
  pilihan = "F14" : Help(admin)
  pilihan = "F16" : save()
  pilihan = "F17" : Exit()

```

	<pre> pilihan = "B02" : kerangajaib() pilihan = "B03" : tictactoe() pilihan = "T01" : <u>if</u> (admin) <u>then</u> change_admin_or_user() <u>else</u> <u>output</u>(hanya_admin) <u>else</u> <u>output</u>("Tidak ada pilihan tersebut. Silakan pilih yang ada di menu.") Wait() menu() </pre>
3.	<p>function cek_role (username : <u>string</u>) -> <u>boolean</u> { Mengembalikan nilai True jika role pengguna adalah admin }</p> <p>KAMUS LOKAL</p> <pre> valid : <u>boolean</u> i : <u>integer</u> </pre> <p>ALGORITMA</p> <pre> valid <- <u>false</u> i <u>traversal</u> [1..Length(data.user)] <u>if</u> (username = data.user[i][1]) <u>and</u> (data.user[i][4] = "admin") <u>then</u> valid <- <u>true</u> -> valid </pre>

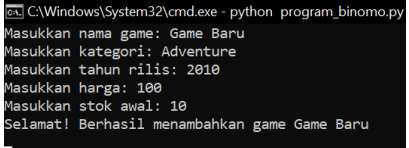
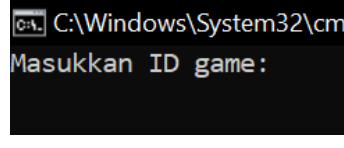
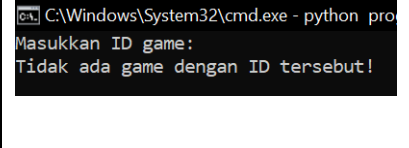
7. SCREENSHOT HASIL PENGUJIAN PROGRAM

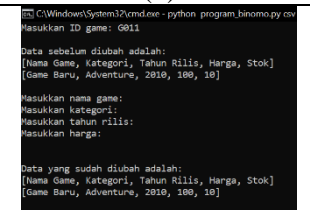
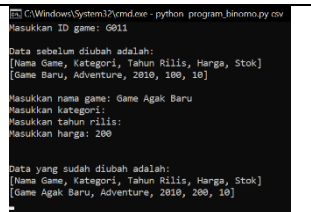
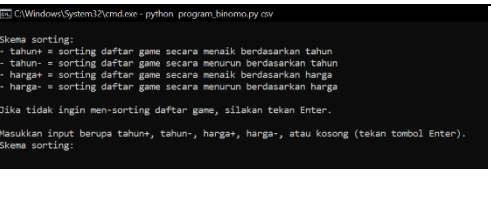
<p>Gambar 3. Pengujian Kerja Fungsi Load (1)</p> 	<p>Gambar 4. Pengujian Fungsi Load (2)</p> 	<p>Gambar 5. Menu Login Awal</p> 
F15 Bila kolom argumen dikosongkan	F15 Bila folder penyimpanan tidak ditemukan	F15 Bila berhasil load

<p>Gambar 6. Pengujian Fungsi Login</p> 	<p>Gambar 7. Menu Program Utama</p> 
F03 jika username atau password yang dimasukkan salah	F03 jika berhasil login


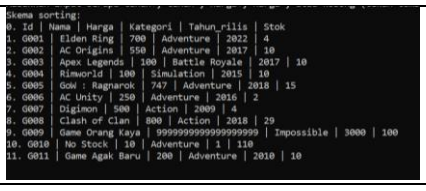
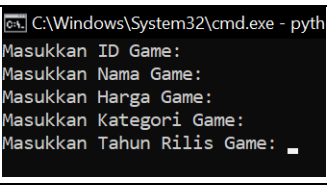
<p>Gambar 8. Kolom Pengisian Fungsi Register</p> 	<p>Gambar 9. Testing Fungsi Register (1)</p> 	<p>Gambar 10. Testing Fungsi Register (2)</p> 
-	F02 jika tidak ada kolom yang diisi	F02 jika kolom username diisi karakter non-valid

<p>Gambar 11. Testing Fungsi Register (3)</p> 	<p>Gambar 12. Kolom Pengisian Fungsi Menambah Game</p> 	<p>Gambar 13. Testing Fungsi Menambah Game (1)</p> 
F02 berhasil register	-	Pesan error F04 bila ada kolom yang tidak diisi

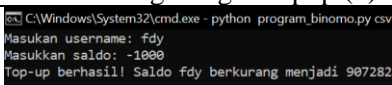

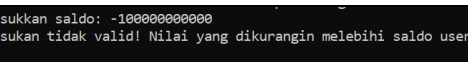
Gambar 14. Testing Fungsi Menambah Game (2)	Gambar 15. Kolom Pengisian Fungsi Mengubah Game	Gambar 16. Testing Fungsi Mengubah Game (1)
 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py Masukkan nama game: Game Baru Masukkan kategori: Adventure Masukkan tahun rilis: 2010 Masukkan harga: 100 Masukkan stok awal: 10 Selamat! Berhasil menambahkan game Game Baru</pre>	 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py Masukkan ID game:</pre>	 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py Masukkan ID game: Tidak ada game dengan ID tersebut!</pre>
Pesan berhasil menambah game	-	Pesan error F06 bila kolom pengisian ID Game kosong atau tidak ditemukan

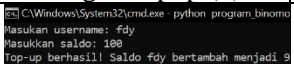
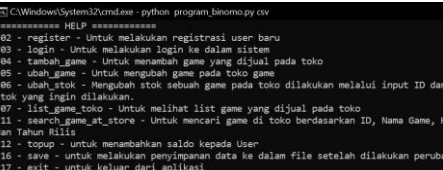
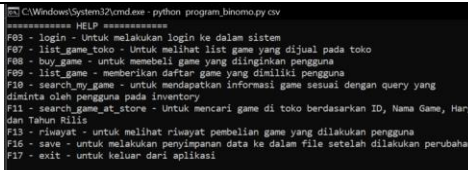
Gambar 17. Testing Fungsi Mengubah Game (2)	Gambar 18. Testing Fungsi Mengubah Game (3)	Gambar 19. Kolom Pengisian Fungsi List Game Toko
 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py Masukkan ID game: G011 Data sebelum diubah adalah: [Nama Game, Kategori, Tahun Rilis, Harga, Stok] [Game Baru, Adventure, 2010, 100, 10] Masukkan nama game: Masukkan kategori: Masukkan tahun rilis: Masukkan harga: Data yang sudah diubah adalah: [Nama Game, Kategori, Tahun Rilis, Harga, Stok] [Game Baru, Adventure, 2010, 100, 10]</pre>	 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py Masukkan ID game: G011 Data sebelum diubah adalah: [Nama Game, Kategori, Tahun Rilis, Harga, Stok] [Game Baru, Adventure, 2010, 100, 10] Masukkan nama game: Game Agak Baru Masukkan kategori: Masukkan tahun rilis: Masukkan harga: 200 Data yang sudah diubah adalah: [Nama Game, Kategori, Tahun Rilis, Harga, Stok] [Game Agak Baru, Adventure, 2010, 200, 10]</pre>	 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py csv Skema sorting: - tahun = sorting daftar game secara menaik berdasarkan tahun - tahun = sorting daftar game secara menurun berdasarkan tahun - harga = sorting daftar game secara menaik berdasarkan harga - harga = sorting daftar game secara menurun berdasarkan harga Jika tidak ingin men-sorting daftar game, silakan tekan Enter. Masukkan input berupa tahun-, tahun-, harga-, harga-, atau kosong (tekan tombol Enter). Skema sorting:</pre>
Kasus F05, tidak ada atribut yang diubah	Pesan berhasil pengubahan atribut game	-

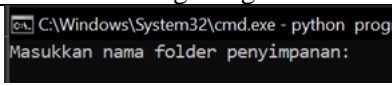
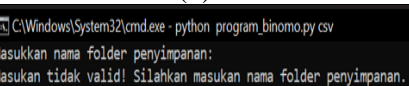
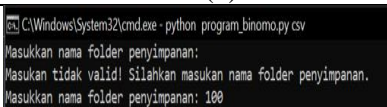
Gambar 20. Testing Fungsi List Game Toko (1)	Gambar 21. Testing Fungsi List Game Toko (2)	Gambar 22. Testing Fungsi List Game Toko (3)
 <pre>Skema sorting: harga- 0. Id Nama Harga Kategori Tahun_rilis Stok 1. G010 No Stock 10 Adventure 1 110 2. G003 Apex Legends 100 Battle Royale 2017 10 3. G004 Rimworld 100 Simulation 2015 10 4. G011 Game Agak Baru 200 Adventure 2010 10 5. G006 AC Unity 250 Adventure 2016 2 6. G007 Digimon 500 Action 2009 4 7. G002 AC Origins 550 Adventure 2017 10 8. G001 Elden Ring 700 Adventure 2022 4 9. G005 GOW : Ragnarok 747 Adventure 2018 15 10. G008 Clash of Clan 800 Action 2018 29 11. G009 Game Orang Kaya 9999999999999999999 Impossible 3000 100</pre>	 <pre>Skema sorting: harga- 0. Id Nama Harga Kategori Tahun_rilis Stok 1. G009 Game Orang Kaya 9999999999999999999 Impossible 3000 100 2. G008 Clash of Clan 800 Action 2018 29 3. G005 GOW : Ragnarok 747 Adventure 2018 15 4. G001 Elden Ring 700 Adventure 2022 4 5. G002 AC Origins 550 Adventure 2017 10 6. G007 Digimon 500 Action 2009 4 7. G006 AC Unity 250 Adventure 2016 2 8. G011 Game Agak Baru 200 Adventure 2010 10 9. G004 Rimworld 100 Simulation 2015 10 10. G003 Apex Legends 100 Battle Royale 2017 10 11. G010 No Stock 10 Adventure 1 110</pre>	 <pre>Skema sorting: tahun- 0. Id Nama Harga Kategori Tahun_rilis Stok 1. G010 No Stock 10 Adventure 1 110 2. G007 Digimon 500 Action 2009 4 3. G011 Game Agak Baru 200 Adventure 2010 10 4. G004 Rimworld 100 Simulation 2015 10 5. G006 AC Unity 250 Adventure 2016 2 6. G003 Apex Legends 100 Battle Royale 2017 10 7. G002 AC Origins 550 Adventure 2017 10 8. G008 Clash of Clan 800 Action 2018 29 9. G005 GOW : Ragnarok 747 Adventure 2018 15 10. G001 Elden Ring 700 Adventure 2022 4 11. G009 Game Orang Kaya 9999999999999999999 Impossible 3000 100</pre>
Fungsi F07 Menunjukkan game toko secara dengan urutan harga menaik	Fungsi F07 Menunjukkan game toko secara dengan urutan harga menurun	Fungsi F07 Menunjukkan game toko secara dengan urutan tahun menaik

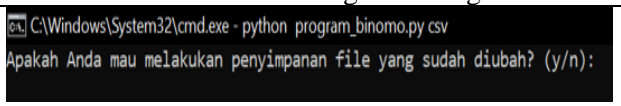
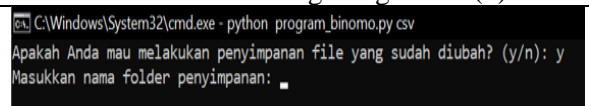
Gambar 23. Testing Fungsi List Game Toko (4)	Gambar 24. Testing Fungsi List Game Toko (4)	Gambar 25. Kolom Pengisian Fungsi Cari Game Toko
 <pre>Skema sorting: tahun- 0. Id Nama Harga Kategori Tahun_rilis Stok 1. G009 Game Orang Kaya 9999999999999999999 Impossible 3000 100 2. G001 Elden Ring 700 Adventure 2022 4 3. G005 GOW : Ragnarok 747 Adventure 2018 15 4. G008 Clash of Clan 800 Action 2018 29 5. G002 AC Origins 550 Adventure 2017 10 6. G003 Apex Legends 100 Battle Royale 2017 10 7. G006 AC Unity 250 Adventure 2016 2 8. G004 Rimworld 100 Simulation 2015 10 9. G007 Digimon 500 Action 2009 4 10. G010 No Stock 10 Adventure 1 110</pre>	 <pre>Skema sorting: 0. Id Nama Harga Kategori Tahun_rilis Stok 1. G001 Elden Ring 700 Adventure 2022 4 2. G002 AC Origins 550 Adventure 2017 10 3. G003 Apex Legends 100 Battle Royale 2017 10 4. G004 Rimworld 100 Simulation 2015 10 5. G005 GOW : Ragnarok 747 Adventure 2018 15 6. G006 AC Unity 250 Adventure 2016 2 7. G007 Digimon 500 Action 2009 4 8. G008 Clash of Clan 800 Action 2018 29 9. G009 Game Orang kaya 9999999999999999999 Impossible 3000 100 10. G010 No Stock 10 Adventure 1 110 11. G011 Game Agak Baru 200 Adventure 2010 10</pre>	 <pre>C:\Windows\System32\cmd.exe - python program_binomo.py Masukkan ID Game: Masukkan Nama Game: Masukkan Harga Game: Masukkan Kategori Game: Masukkan Tahun Rilis Game:</pre>
Fungsi F07 Menunjukkan game toko secara dengan urutan tahun menurun	Fungsi F07 Menunjukkan game toko secara dengan game id terurut	-

Gambar 26. Testing Fungsi Cari Game Toko (1)	Gambar 27. Testing Fungsi Cari Game Toko (2)	Gambar 28. Testing Fungsi Cari Game Toko (3)
 <p>Masukkan ID Game: G7idakQA Masukkan Nama Game: Masukkan Harga Game: Masukkan Kategori Game: Masukkan Tahun Rilis Game: Daftar game pada toko yang memenuhi kriteria: Tidak ada game pada inventory-mu yang memenuhi kriteria</p>	 <p>Masukkan ID Game: Masukkan Nama Game: Masukkan Harga Game: Masukkan Kategori Game: Masukkan Tahun Rilis Game: Daftar game pada toko yang memenuhi kriteria: 0. Id Nama Kategori Tahun_rilis Harga Stok 1. G001 Elden Ring Adventure 2022 780 4 2. G002 AC Origins Adventure 2017 550 10 3. G003 Apex Legends Battle Royale 2017 100 10 4. G004 Rimworld Simulation 2015 100 10 5. G005 Goli : Ragnarok Adventure 2018 747 15 6. G006 AC Unity Adventure 2016 250 2 7. G007 Digimon Action 2009 500 4 8. G008 Clash of Clan Action 2013 800 29 9. G009 Game Orang Kaya Impossible 3000 99999999999999999999 100 10. G010 No Stock Adventure 1 10 110 11. G011 Game Agak Baru Adventure 2010 200 10</p>	 <p>Masukkan ID Game: Masukkan Nama Game: Masukkan Harga Game: Masukkan Kategori Game: Masukkan Tahun Rilis Game: 2017 Daftar game pada toko yang memenuhi kriteria: 0. Id Nama Kategori Tahun_rilis Harga Stok 1. G002 AC Origins Adventure 2017 550 10 2. G003 Apex Legends Battle Royale 2017 100 10</p>
Kasus F11, Tidak ditemukan game dengan atribut yang diisi	Kasus F11, Seluruh kolom dikosongkan; ditampilkan seluruh game di toko	Kasus F11, Ditemukan seluruh game dengan atribut yang diisi

Gambar 29. Kolom Pengisian dan Testing Fungsi Topup (1)	Gambar 30. Testing Fungsi Topup (2)	Gambar 31. Testing Fungsi Topup (3)
 <p>Masukkan username: fdy Masukkan saldo: -1000 Top-up berhasil! Saldo fdy berkurang menjadi 907282</p>	 <p>Masukkan username: user_tidak_Ada Masukkan saldo: 10000 Username "user_tidak_Ada" tidak ditemukan</p>	 <p>Masukkan saldo: -1000000000000 Masukan tidak valid! Nilai yang dikurangkan melebihi saldo user</p>
Kasus F12, Saldo berhasil dikurangkan	Kasus F12, Username tidak ditemukan	Kasus F12, Saldo tidak bisa dikurangi karena melebihi saldo user

Gambar 32. Testing Fungsi Topup (4)	Gambar 33. Testing Fungsi Help Untuk Admin	Gambar 34. Testing Fungsi Help Untuk User
 <p>Masukkan username: fdy Masukkan saldo: 100 Top-up berhasil! Saldo fdy bertambah menjadi 907382</p>	 <p>===== HELP ===== F02 - register - Untuk melakukan registrasi user baru F03 - login - Untuk melakukan login ke dalam sistem F07 - list_game_toko - Untuk melihat list game yang dijual pada toko F08 - tambah_game - Untuk menambah game yang dijual pada toko F05 - ubah_game - Untuk mengubah game pada toko game F06 - ubah_stok - Mengubah stok sebuah game pada toko dilakukan melalui input ID dan stok yang ingin dilakukan. F07 - list_game_toko - Untuk melihat list game yang dijual pada toko F11 - search_game_at_store - Untuk mencari game di toko berdasarkan ID, Nama Game, dan Tahun Rilis F12 - topup - untuk menambahkan saldo kepada User F16 - save - untuk melakukan penyimpanan data ke dalam file setelah dilakukan perubahan F17 - exit - untuk keluar dari aplikasi</p>	 <p>===== HELP ===== F02 - login - Untuk melakukan login ke dalam sistem F07 - list_game_toko - Untuk melihat list game yang dijual pada toko F08 - buy_game - untuk membeli game yang diinginkan pengguna F09 - list_game - memberikan daftar game yang dimiliki pengguna F10 - search_my_game - untuk mendapatkan informasi game sesuai dengan query yang diminta oleh pengguna pada inventory F11 - search_game_at_store - Untuk mencari game di toko berdasarkan ID, Nama Game, dan Tahun Rilis F13 - riwayat - untuk melihat riwayat pembelian game yang dilakukan pengguna F16 - save - untuk melakukan penyimpanan data ke dalam file setelah dilakukan perubahan F17 - exit - untuk keluar dari aplikasi</p>
Pesan berhasil menambah game	-	-

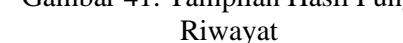
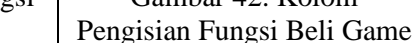
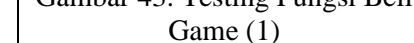
Gambar 35. Kolom Pengisian dan Testing Fungsi Save	Gambar 36. Testing Fungsi Save (1)	Gambar 37. Testing Fungsi Save (2)
 <p>Masukkan nama folder penyimpanan: Masukkan nama folder penyimpanan: C:\Windows\System32 Masukan tidak valid! Silahkan masukan nama folder penyimpanan. Masukkan nama folder penyimpanan: 100</p>	 <p>Masukkan nama folder penyimpanan: Masukan tidak valid! Silahkan masukan nama folder penyimpanan.</p>	 <p>Masukkan nama folder penyimpanan: Masukan tidak valid! Silahkan masukan nama folder penyimpanan. Masukkan nama folder penyimpanan: 100</p>
-	Kasus F16, Kolom pengisian tidak diisi	Kasus F16, File berhasil di save

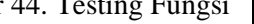
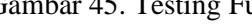

Gambar 38. Kolom Pengisian Fungsi Exit	Gambar 39. Testing Fungsi Exit (1)
 <p>Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n):</p>	 <p>Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n): y Masukkan nama folder penyimpanan:</p>
-	Kasus F17, Memilih ingin save file

Gambar 40. Testing Fungsi Exit (2)

```
C:\Windows\System32\cmd.exe
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n): n
D:\code\TPB\DASPRO\TUBES>
```



Berhasil exit program

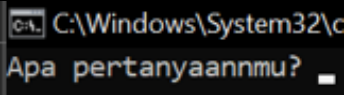
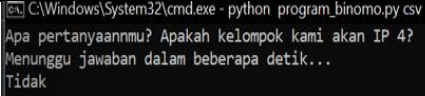
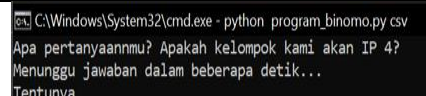
<p align="center">Gambar 41. Tampilan Hasil Fungsi Riwayat</p> 	<p align="center">Gambar 42. Kolom Pengisian Fungsi Beli Game</p> 	<p align="center">Gambar 43. Testing Fungsi Beli Game (1)</p> 
<p align="center">-</p>	<p align="center">-</p>	<p>Kasus F08, Gagal beli : Game tidak ditemukan atau kolom kosong</p>


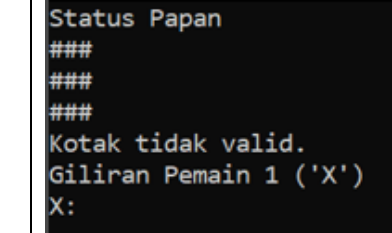
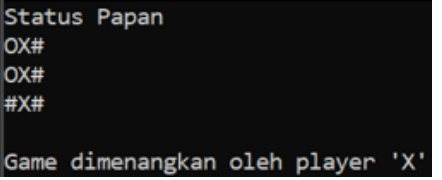
Gambar 44. Testing Fungsi Beli Game (2)	Gambar 45. Testing Fungsi Beli Game (3)	Gambar 46. Testing Fungsi Beli Game (4)
 <p>C:\Windows\System32\cmd.exe - python Masukkan ID game: G001 Stok Game tersebut sedang habis!</p>	 <p>C:\Windows\System32\cmd.exe - python Masukkan ID game: G001 Saldo anda tidak cukup untuk membeli Game tersebut!</p>	 <p>C:\Windows\System32\cmd.exe - python Masukkan ID game: G003 Game "Apex Legends" berhasil dibeli!</p>
Kasus F08, Gagal beli : Stok game sedang habis	Kasus F08, Gagal beli : Saldo user tidak cukup untuk membeli game	Kasus F08, Game berhasil dibeli

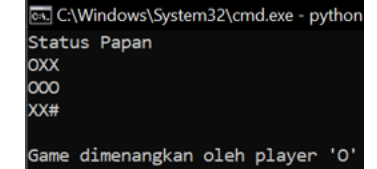
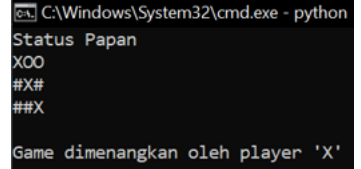
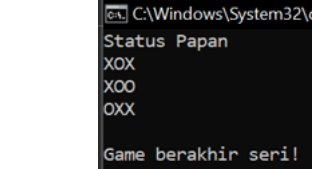
The figure consists of three screenshots from a Windows command prompt running a Python program.

- Gambar 47. Tampilan Hasil Fungsi List Game:** Shows the output of the 'Daftar Game' function, displaying a list of 6 games with their ID, Name, Category, Release Year, and Stock.
- Gambar 48. Kolom Pengisian Fungsi Cari Game Saya:** Shows the input fields for the 'Cari Game' function, including fields for 'Masukkan ID Game:', 'Masukkan Nama Game:', 'Masukkan Harga Game:', 'Masukkan Kategori Game:', and 'Masukkan Tahun Rilis Game:'.
- Gambar 49. Testing Fungsi Cari Game Saya (1):** Shows the output of the 'Daftar game pada toko yang memenuhi kriteria:' function, displaying a list of 10 games with their ID, Name, Category, Release Year, and Stock.

<p>Gambar 50. Testing Fungsi Cari Game Saya (2)</p>  <pre> C:\Windows\System32\cmd.exe - python program_binomo.py csv Masukkan ID Game: Gametidakada Masukkan Nama Game: Masukkan Harga Game: Masukkan Kategori Game: Masukkan Tahun Rilis Game: Daftar game pada toko yang memenuhi kriteria: Tidak ada game pada inventory-mu yang memenuhi kriteria </pre>	<p>Gambar 51. Testing Fungsi Cari Game Saya (3)</p>  <pre> C:\Windows\System32\cmd.exe - python program_binomo.py csv Masukkan ID Game: Masukkan Nama Game: Masukkan Harga Game: Masukkan Kategori Game: Adventure Masukkan Tahun Rilis Game: Daftar game pada toko yang memenuhi kriteria: 0. Id Nama Kategori Tahun_rilis Harga Stok 1. G001 Elden Ring Adventure 2022 700 0 2. G005 Ragnarok Adventure 2018 747 15 3. G002 AC Origins Adventure 2017 550 10 4. G006 AC Unity Adventure 2016 250 2 5. G010 No Stock Adventure 1 10 10 </pre>
<p>Kasus F10, Tidak ada game yang sesuai dengan atribut isian</p>	<p>Kasus F10, Ditemukan semua game dengan atribut isian</p>

Gambar 52. Kolom Pengisian Fungsi Magic Conch Shell	Gambar 53. Testing Fungsi Fungsi Magic Conch Shell (1)	Gambar 54. Testing Fungsi Fungsi Magic Conch Shell (2)
		
-	-	-

Gambar 55. Tampilan Permainan Fungsi Tic Tac Toe	Gambar 56. Testing Permainan Fungsi Tic Tac Toe (1)	Gambar 57. Testing Permainan Fungsi Tic Tac Toe (2)
		
-	Pengisian kotak tidak valid karena <i>out of bounds</i> atau petak sudah ditempati	Kasus B03, Permainan dimenangkan dengan cara vertikal

Gambar 58. Testing Permainan Fungsi Tic Tac Toe (3)	Gambar 59. Testing Permainan Fungsi Tic Tac Toe (4)	Gambar 60. Testing Permainan Fungsi Tic Tac Toe (5)
		
Kasus B03, Permainan dimenangkan dengan cara horizontal	Kasus B03, Permainan dimenangkan dengan cara diagonal	Kasus B03, Permainan berakhir seri

LAMPIRAN

Gambar 61. Form Asistensi 1

Form MoM Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2021/2022

Nomor Asistensi	:	01
No. Kelompok/Kelas	:	10/08
Tanggal asistensi	:	Senin, 11 April 2022
Anggota kelompok	:	NIM / Nama (Hanya yang Hadir)
1	:	16521034/Satria Octavianus Nababan
2	:	16521115/Fawwaz Abrial Saffa
3	:	16521124/Frendy Sanusi
4	:	16521160/Adinda Siti Fathonah
5	:	
6	:	
Asisten pembimbing	:	NIM / Nama 13519140/Fabian Savero Diaz Pranoto

Catatan Asistensi:

Rangkuman Diskusi
<p>Q : Untuk file kepemilikan atribut "user_id", bagaimana cara menyimpannya? A : Disatukan saja. Misalkan ada dua user yang mempunyai game yang sama, nanti di baris satu bisa diisi G001 dan user_id-nya, lalu di baris dua diisi G001 lagi, tetapi dengan user_id yang berbeda.</p> <p>Q : Buat H02, karena akses awalnya hanya admin yang bisa, berarti jika dimulai dari file csv kosong tidak bisa, ya? Apakah harus diisi admin dulu? A : Iya, kasih masukan admin saja di file csv-nya, misalnya admin1.</p> <p>Q : Apakah masing-masing fungsi dijadikan per-module atau boleh digabung? A : Kalau mau digabung boleh saja. Biasanya yang fungsionalitasnya mirip boleh disatu-satun, misalnya login dan register, load dan save. Tidak harus satu module berisi satu fungsi, disatukan saja fungsi-fungsi yang ada di satu module masih berimbang.</p> <p>Q : Buat riwayat.csv, apakah per-user harus dijadikan satu file atau digabung dengan user lain? A : Digabungkan saja karena ada kolom user_id sebagai pembeda.</p> <p>Q : Jika beli game masuk di file riwayat, apakah tahun belinya diisi 2022 saja atau bagaimana? A : Terserah kalian, tidak ada batasan untuk itu.</p>
Tindak Lanjut
<ol style="list-style-type: none"> 1. Cek QnA saja buat mengecek fungsi-fungsi apa yang boleh dipakai dan yang tidak. 2. Boleh chat asisten melalui LINE atau QnA jika masih ada yang bingung. 3. Pada asistensi 2 nanti harus sudah ada progress yang dilaporkan.

Gambar 62. Form Asistensi 2

Form MoM Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2021/2022

Nomor Asistensi	:	02
No. Kelompok/Kelas	:	10/08
Tanggal asistensi	:	Rabu, 20 April 2022
Anggota kelompok	:	NIM / Nama (Hanya yang Hadir)
1	:	16521034/Satria Octavianus Nababan
2	:	16521115/Fawwaz Abrial Saffa
3	:	16521124/Frendy Sanusi
4	:	16521160/Adinda Siti Fathonah
5	:	
6	:	
Asisten pembimbing	:	NIM / Nama 13519140/Fabian Savero Diaz Pranoto

Catatan Asistensi:

Rangkuman Diskusi
<ol style="list-style-type: none"> 1. Desain command boleh disatukan dengan desain kamus notasi algoritmik pada pembuatan laporan. 2. Bab dekomposisi pada laporan dibuat dalam bentuk flowchart yang menunjukkan alur dari dekomposisi program. 3. Bab spesifikasi pada laporan ditulis dalam notasi algoritmik. Fungsi yang dibuat notasi algoritmiknya hanya fungsi-fungsi pada spesifikasi wajib, fungsi pelengkap yang bukan spesifikasi wajib boleh dibuat boleh tidak.
Tindak Lanjut
<ol style="list-style-type: none"> 1. Laporannya diselesaikan. 2. Video demo hanya memuat hal-hal yang penting saja.

Gambar 63. Dokumentasi

