

PROGRAMMER TANPA CODING



PROGRAM STUDI
SISTEM INFORMASI
• SISTEM INFORMASI • E-COMMERCE
• GAME TECHNOLOGY • AKUNTANSI-SI
SIEGA

**Sammuel Matthew G.
Marcello Jovan C.K., dkk**

Universitas Katolik Soegijapranata

Tim Penulis

- Samuel Matthew Gunawan
- Marcello Jovan Christopher Kalbuadi
- Kevin Wijaya
- Roberto Antonio Mozart Udju
- Kristoforus Thoeirick Dimasaputra
- Enrico Natanael Nugroho
- Adinata Ekwan Januar Pratama
- Justine Hezekiel Ramli
- Benedictus Brian Renaldi Darma
- Renaldo Haryono Kusumo



PENERBIT:
Universitas Katolik Soegijapranata

Programmer Tanpa Coding

Oleh:

Samuel Matthew Gunawan, Marcello Jovan Christopher Kalbuadi, dkk

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis dan Penerbit.

©Universitas Katolik Soegijapranata 2022

Desain Sampul : Adinata Ekwan Januar Pratama

Perwajahan Isi : Samuel Matthew Gunawan dan Marcello Jovan Christopher Kalbuadi

Ukuran buku : A4 (21 x 29.7 cm)

Font : Calibri 18

Tanggal Terbit : 20 April 2022

PENERBIT:

Universitas Katolik Soegijapranata

Anggota APPTI No. 003.072.1.1.2019

Anggota IKAPI No 209/ALB/JTE/2021

Jl. Pawiyatan Luhur IV/1 Bendan Duwur Semarang 50234

Telpon (024)8441555 ext. 1409

Website: www.unika.ac.id

Email Penerbit: ebook@unika.ac.id

Katalog Dalam Terbitan:

Prakata

Sebagian besar dari kita sudah pernah mengetahui dan memainkan game namun tidak semua orang mengetahui cara membuat game. Dalam buku ini kita akan melihat game buatan 10 mahasiswa Unika dari jurusan Sistem Informasi. Sesuai dengan judul buku, game-game yang dibuat oleh mahasiswa tidak menggunakan coding sama sekali dengan menggunakan RPG Maker, MIT App Inventor, dan Quickapp Ninja.

Buku ini terdiri dari 3 bagian utama, dimana 7 judul pertama merupakan game yang dibuat menggunakan RPG Maker, judul ke-8 & ke-9 menggunakan MIT App Inventor, dan judul terakhir menggunakan Quickapp Ninja.

Selain dapat melihat hasil dan download game yang sudah dibuat pada setiap akhir bab, Anda juga bisa melihat proses pembuatan game dari awal hingga akhir yang dilengkapi dengan gambar untuk memperjelas tahapan.

Dengan selesainya penulisan buku ini, penulis menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang sudah membantu sehingga buku ini dapat diselesaikan.

Penulis

Kata Pengantar

Pertama-tama kita panjatkan puji dan syukur kepada Tuhan Yang Maha Esa atas terbitnya buku “Programmer Tanpa Coding” yang disusun penuh semangat oleh masing-masing penulisnya dalam rentang waktu 8 (delapan) minggu. Suatu capaian yang membanggakan, terutama bagi para penulis yang baru kuliah di tahun pertama.

Yang menarik, tulisan-tulisan di dalam buku ini tidak sekedar membahas mengenai game yang telah ada, tetapi menciptakannya terlebih dahulu berdasarkan skenario atau gameplay yang diinginkan. Tahapan dalam pembuatan game diceritakan secara terurut dan langkah-langkah yang mudah, bahkan bagi para pemula sekalipun yang tidak memiliki pengalaman coding sebelumnya. Sebab para penulisnya adalah mereka-mereka yang baru kuliah di tahun pertama dimana pembuatan aplikasi atau game umumnya masih menggunakan alat bantu pembuatan tanpa coding.

Beberapa contoh yang dibuat di dalam buku ini dapat dimanfaatkan untuk contoh proyek di semester-semester awal kuliah di jurusan Komputer atau bagi masyarakat awam yang ingin mengembangkan aplikasi atau game sederhana namun mengikuti alur cerita yang dibuat sendiri.

Semoga cara berpikir, usaha dalam menimba pengetahuan dari berbagai literasi, dan keinginan untuk berbagi dari para

penulis kepada sesama yang dituangkan dalam wujud tulisan ini dapat bermanfaat bagi kita semua. Diharapkan dapat menjadi tambahan wawasan yang ringan untuk dibaca dalam melihat ke isi hati setiap penulisnya.

Prof. Dr. Ridwan Sanjaya

Daftar Isi

Prakata	iii
Kata Pengantar	iv
Daftar Isi	vi
GEMAR MENGHITUNG	1
Game Engine	2
Mengunduh Game Engine	3
Spesifikasi Game Engine	4
Instalasi Game Engine	5
TANTANGAN GAME	8
Mekanisme Game	8
Game	9
Kontrol	9
KARAKTER DALAM GAME	9
TAHAPAN MEMBUAT GAME	14
Membuat prolog	16
Membuat peta dan NPC	18
Link download game	32
PILIH JALAN YANG BENAR	33
LATAR BELAKANG GAME	36
Jalan Cerita	37
TANTANGAN GAME	38
KARAKTER PEMAIN	40
TAHAPAN PEMBUATAN GAME	41
Scripting	52

Ekspor Game	69
Link Video Gameplay Dan Link Download Game	71
Kota Yang Terbangkalai	72
Tampilan Keseluruhan Map	72
Memulai RPG Maker MV	74
Meng-install Plugin	75
Mengatur Database	80
Membuat Map	100
Membuat Event	124
Membuat Lampu	124
Membuat Pintu Terbuka	126
Membuat Pintu Terkunci	128
Membuat Pintu Bersyarat	132
Membuat Event Keluar Map	149
Membuat Event Utama	152
Adegan Akhir	203
Membuat Credit	207
Membuat Intro	211
Membuat Event Tambahan	214
Hasil Akhir	218
CARI JALAN KELUAR	219
Analisa Game	220
Mission Statement	220
Genre	220
Platform	220
Target Audience	221
Gambaran Umum Gameplay	221
Pengalaman Pemain	221

Panduan Gameplay	222
Mekanik Gameplay	222
Karakter	223
Lise (MC)	224
Info Manusia	224
<i>Parasitic Worm</i>	226
Info Parasit	227
<i>Zombie I (ZI)</i>	228
<i>Zombie II (ZII)</i>	229
Info <i>Zombie</i>	230
Desain Level	232
Level 1	233
Level 2	234
Level 3	235
Level 4	236
Level 5	237
Jalan cerita	238
Interogasi dan Edukasi	242
Tantangan	243
Karakter	243
Map	247
Tahap Pembuatan	250
Link game	309
Petualangan Pelajar	310
TANTANGAN	310
KARAKTER GAME	311
Peta	312
TAHAPAN PEMBUATAN GAME	314

NPC (Non-Playable Character)	320
Contoh Pertanyaan	327
Teleport	329
Dialog jika pemain menang atau kalah dalam bertarung	332
Cara ekspor game	337
Dragon Hunter	340
Musik Mini Quiz	357
Hasil jadi	381
Membuat Game Menemukan Harta Karun	382
Pengenalan MIT	382
Membuat Game	386
Link:	404
Tantangan Tebak Kata Dan Gambar	405
Bab Isi	406
Link Game:	414
Daftar Pustaka	428
Biografi	429

GEMAR MENGHITUNG

Roberto Antonio Mozart Udju

Perkembangan teknologi yang sangat pesat menjadi pemicu dari munculnya banyak teknologi baru. Khususnya di dunia *game* atau *developer game*. Jadi, tidak heran banyak sekali *game* yang bermunculan setiap tahun bahkan setiap bulannya dari berbagai macam *developer* atau *publisher*. Dalam *men-develop* sebuah *game* kita juga membutuhkan *engine* untuk membuat *game*. Berbagai macam *engine* yang tersedia sekarang mulai dari yang menggunakan koding sampai yang tanpa koding.

RPG Maker contoh dari *engine* membuat *game* tanpa menggunakan kodingan. *Engine* ini adalah *game engine* yang cocok untuk pemula selain tanpa koding, *engine* ini juga mudah untuk dimengerti, tidak hanya mudah *game* yang bisa di buat juga bermacam-macam. *Game* dengan *genre action*, petualangan dan edukasi menjadi *genre game* yang gemari banyak orang. Ada banyak hal yang dapat diambil dari *game-game* tersebut, bukan sekedar seru untuk dimainkan tetapi ada pengetahuan yang dapat diambil.

Game Engine

Dalam proses pembuatan *game* pasti memerlukan alat atau mesin yang biasa di sebut *game engine*. *Game engine* sendiri sudah berbagai macam, tidak bisa dipungkiri *game engine* sendiri mengalami banyak perkembangan seiring dengan berjalannya waktu ke waktu.

Ada beberapa contoh *game engine*:

1. Unity



LOGO UNITY

Gambar 1.1. (*Logo Unity*, n.d.)

Unity adalah *game engine* yang dibuat oleh Unity Technologies. Unity Technologies sendiri didirikan pada tahun 2004. Pendiri Unity Technologies adalah David Helgason, Nicholas Francis, dan Joachim Ante. Unity adalah salah satu mesin pembuatan *game* yang mudah dipahami dan termasuk lengkap dengan fitur yang berbagai macam yang ada dalam Unity. Pemrograman yang digunakan unity adalah bahasa

pemrograman *C#*. Meskipun mudah digunakan dan dipahami bukan berarti *game engine* satu ini memiliki fitur yang sedikit, bahkan sebaliknya unity memiliki fitur yang beragam.

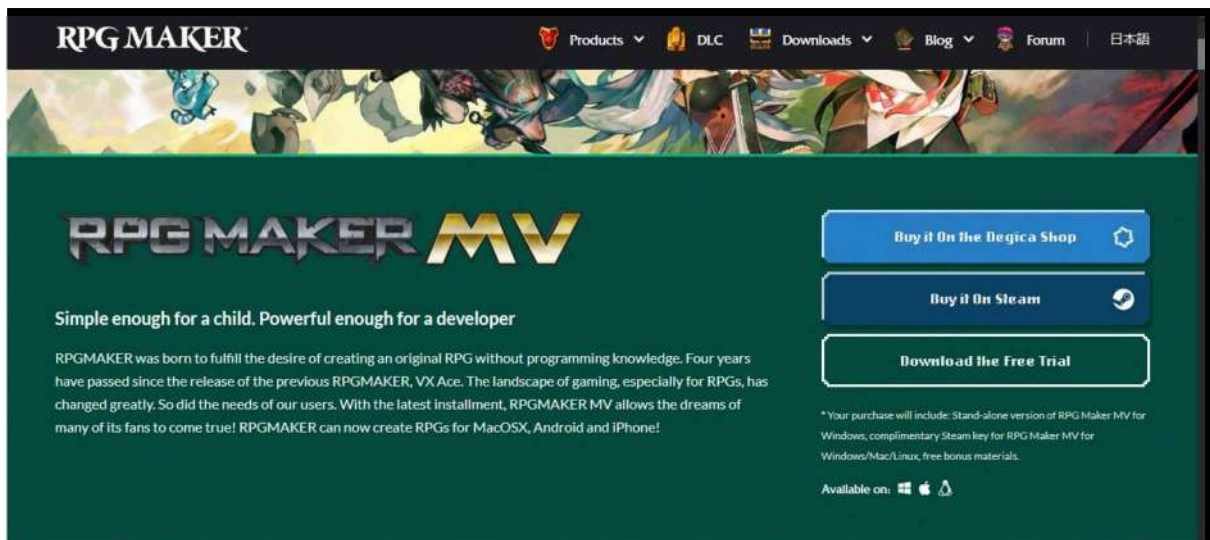
Unity sendiri dapat membuat *game 2D, 3D, virtual reality (VR)*, dan *augmented reality*. Unity dapat beroperasi pada Windows dan Mac Os permainan yang dapat dibuat hampir ada di semua *platform*.

2. RPG Maker MV

RPG Maker MV merupakan salah satu dari *engine* RPG Maker. *Game engine* ini dirancang untuk membuat *game* yang bergenre *role playing game RPG*, *Game engine* ini terbilang sangat mudah dalam pengoperasiannya. Mengapa terbilang sangat mudah? Karena di *game engine* yang satu ini tidak memiliki fitur koding-mengkode. Meskipun tanpa koding anda dapat membuat *game 2D RPG*, yang menarik dengan *game engine* satu ini. *Game engine* ini sudah *support* untuk membuat *game-game android/mobile game*.

Mengunduh Game Engine

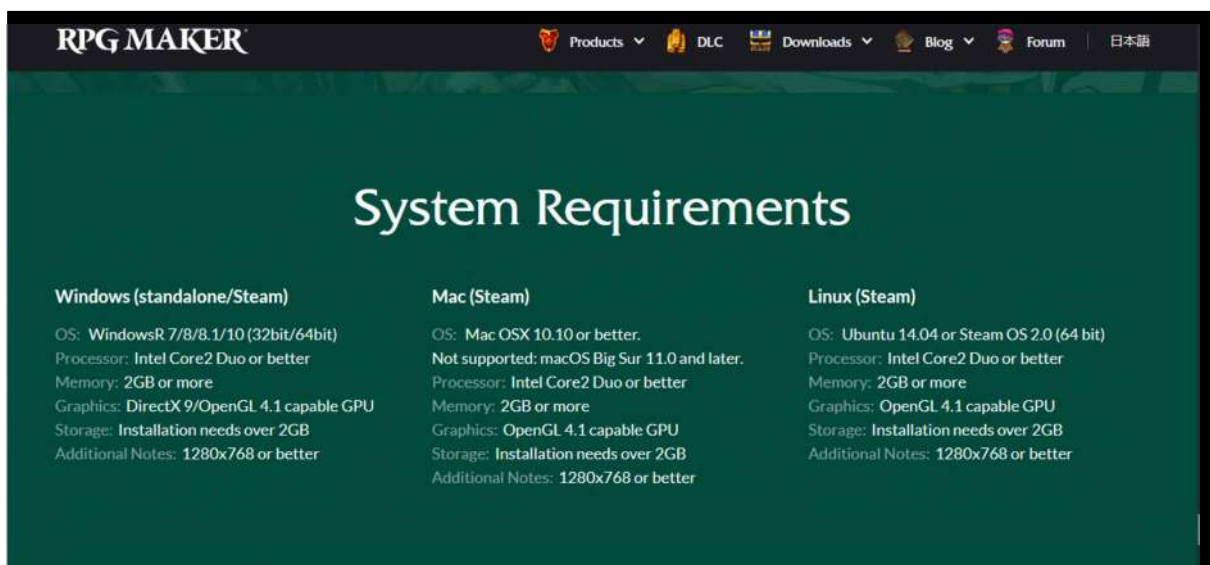
Sebelum menginstal *game engine* (RPG Maker MV), anda harus mengunduhnya terlebih dahulu. Anda dapat mengunduhnya di [www.rpgmakerweb.com](http://www.rpgmakerweb.com/downloads) /downloads. (*Download RPG Maker MV*, n.d.)



Gambar 1.2. Tempat mengunduh RPG Maker MV

Spesifikasi Game Engine

Adapun spesifikasi yang diperlukan dalam mengunduh RPG Maker MV. (*Spesifikasi RPG Maker MV, n.d.*)



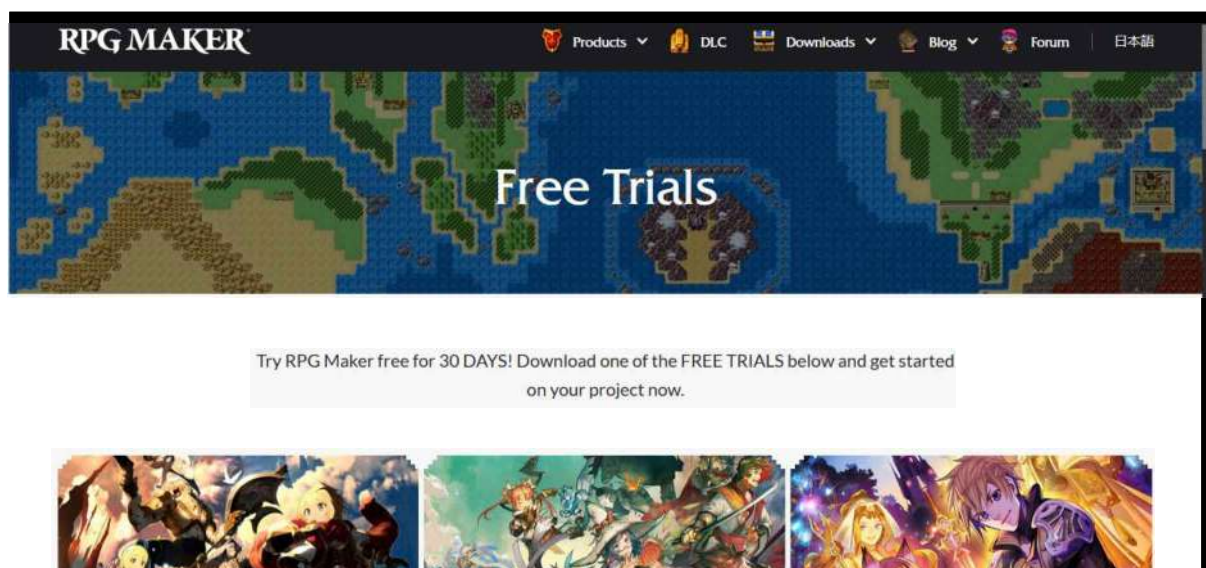
Gambar 1.3. Spesifikasi *device* untuk mengunduh RPG Maker MV

Instalasi Game Engine

Game engine yang digunakan untuk membuat *game* “GEMAR MENGHITUNG” adalah RPG Maker MV, anda bisa mengunduhnya di Steam, dan *web* RPG Maker MV (Berbayar). Jika ingin gratis anda dapat mengunduh versi *trial*-nya saja.



Gambar 1.4. RPG Maker MV berbayar



Gambar 1.5. *Free trials* RPG Maker MV

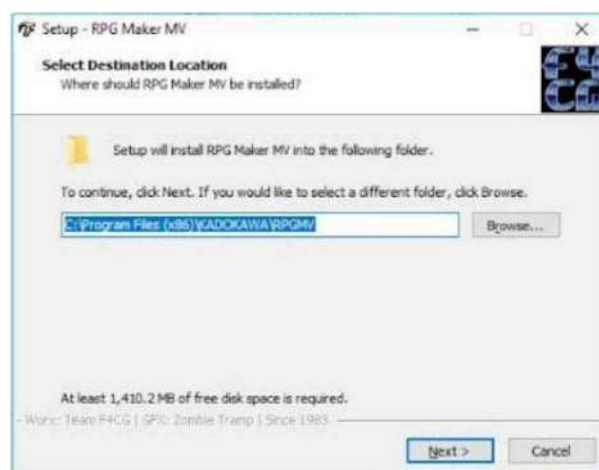
Berikut langkah-langkah meng-*install* RPG Maker MV:

1. Pastikan sudah mempunyai *installer* RPG Maker MV, Jika belum diunduh dulu.
2. Klik kanan *Installer* lalu, *Run As administrator* RPG Maker *installer*-nya.
3. Setelah menjalankan *installer*, maka akan muncul untuk pemilihan bahasa yang dipakai.



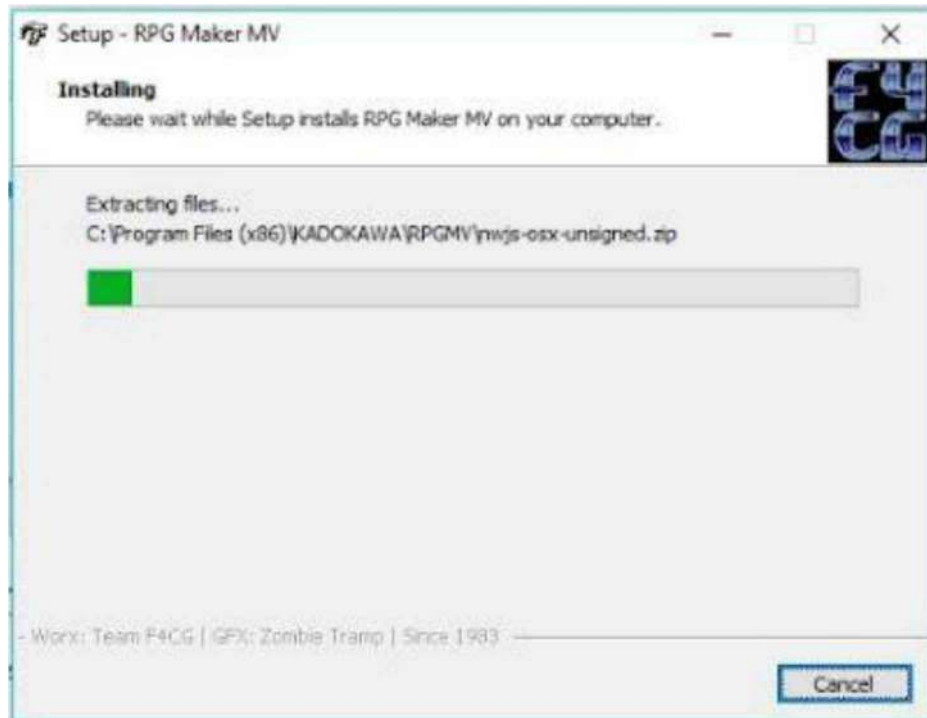
Gambar 1.6. Pemilihan Bahasa di RPG Maker MV

4. Setelah itu memilih tempat penyimpanan, untuk menyimpan *folder* RPG Maker MV.



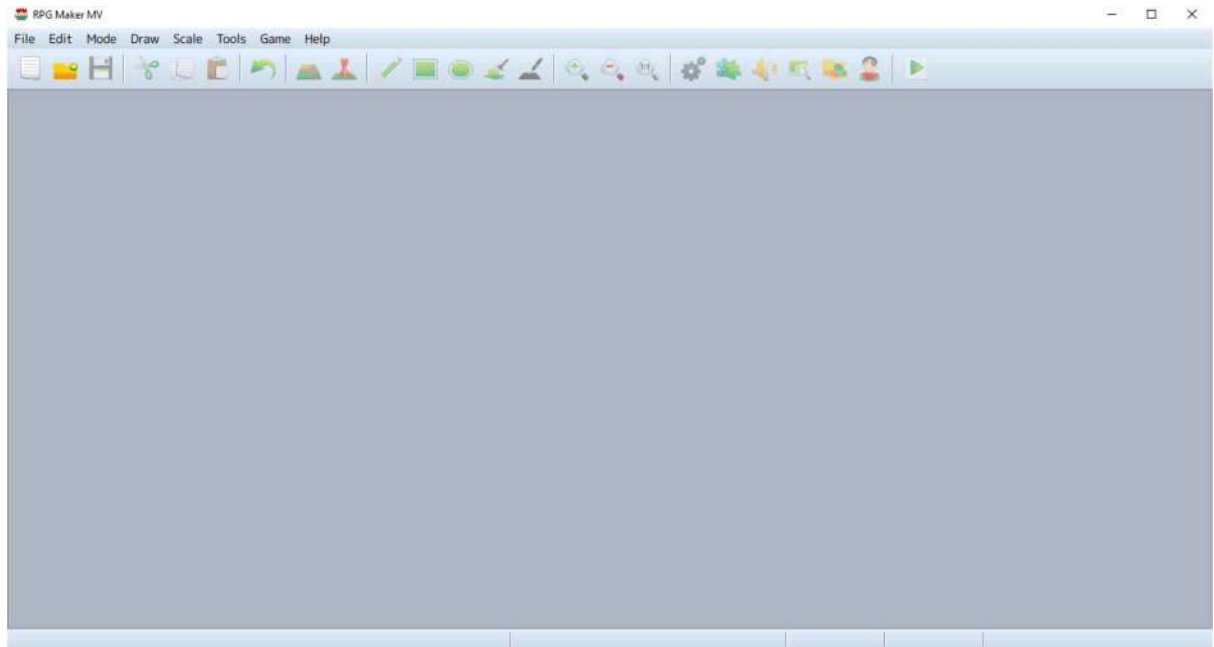
Gambar 1.7. Pemilihan tempat penyimpanan RPG Maker MV

5. Menunggu hingga prosesnya selesai.



Gambar 1.8. Menunggu proses *installing* selesai

6. Jika sudah selesai, buka RPG Maker MV yang sudah di *install* maka masuk tampilan sama seperti dibawah ini.



Gambar 1.9. Tampilan awal RPG Maker MV

TANTANGAN GAME

Tantangan yang ada dalam *game* “GEMAR MENGHITUNG” adalah soal matematika dasar yang diberikan oleh *NPC* (Guru) yang ada di dalam *game*.

Mekanisme Game

Pemain akan menjalankan karakter utama, Pemain akan diberikan soal yang berupa matematika dasar oleh *NPC-NPC* guru yang ada di dalam *game*. Apabila pemain melakukan kesalahan dalam menjawab soal yang diberikan akan dinyatakan jawabannya salah dan harus mengulang dari awal permainan, dalam *game* ini tidak memiliki *checkpoint*. Mengapa *game* ini tidak memiliki *checkpoint*, karena soal atau

pertanyaan yang diberikan terbilang mudah, karena soal yang diberikan sekedar matematika dasar.

Game

Game “GEMAR MENGHITUNG”, adalah sebuah *game* yang bergenre *education-action-puzzle*. *Game* ini ditujukan untuk usia 7 tahun keatas, dalam *game* ini kita akan mengerjakan soal matematika dasar (Penjumlahan, pengurangan, perkalian, pembagian).

Kontrol

Kontrol dalam *game* cukup mudah karena cuman menggerakkan karakter kiri, kanan, atas, bawah, menggunakan tombol anak panah pada *keyboard*. Menggunakan tombol w, a, s, d. Dan dapat menggunakan klik kiri dan kanan pada *mouse*.

KARAKTER DALAM GAME

Dalam *game* ini memiliki memiliki beberapa karakter. Untuk karakter utama adalah Asep, dan Beberapa *NPC* seperti guru dan murid lainnya.

Karakter utama:



Gambar 1.10. Asep

Karakter guru:



Gambar 1.11. Guru kelas penjumlahan



Gambar 1.12. Guru kelas pengurangan



Gambar 1.13. Guru kelas pembagian



Gambar 1.14. Guru kelas perkalian

Karakter murid-murid:



Gambar 1.15. *NPC* murid kelas penjumlahan



Gambar 1.16. *NPC* murid kelas penjumlahan



Gambar 1.17. *NPC* murid kelas penjumlahan



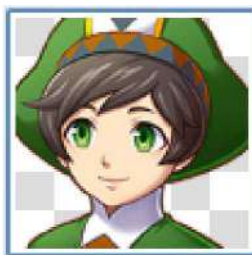
Gambar 1.18. *NPC* murid kelas pembagian



Gambar 1.19. *NPC* murid kelas pengurangan



Gambar 1.20. *NPC* murid pengurangan



Gambar 1.21. *NPC* murid kelas perkalian



Gambar 1.22. *NPC* murid kelas perkalian



Gambar 1.23. *NPC* murid perkalian



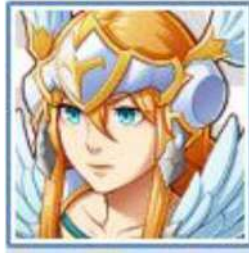
Gambar 1.24. *NPC* murid di halaman sekolah



Gambar 1.25. *NPC* murid di halaman sekolah



Gambar 1.26. *NPC* murid di halaman sekolah



Gambar 1.27. *NPC* murid di halaman sekolah



Gambar 1.28. *NPC* murid di halaman sekolah



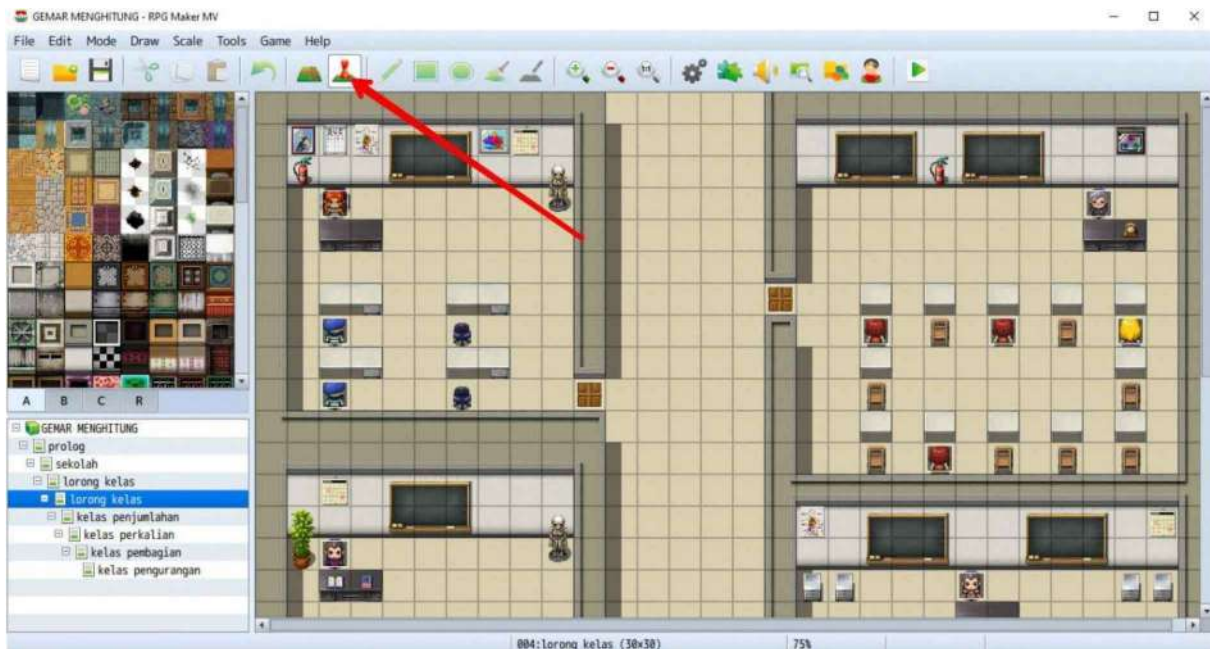
Gambar 1.29. *NPC* murid di halaman sekolah

TAHAPAN MEMBUAT GAME

Sebelum anda membuat *game* di RPG Maker MV. Anda harus mengetahui 2 fitur penting, yaitu *event* dan *map*.

Event

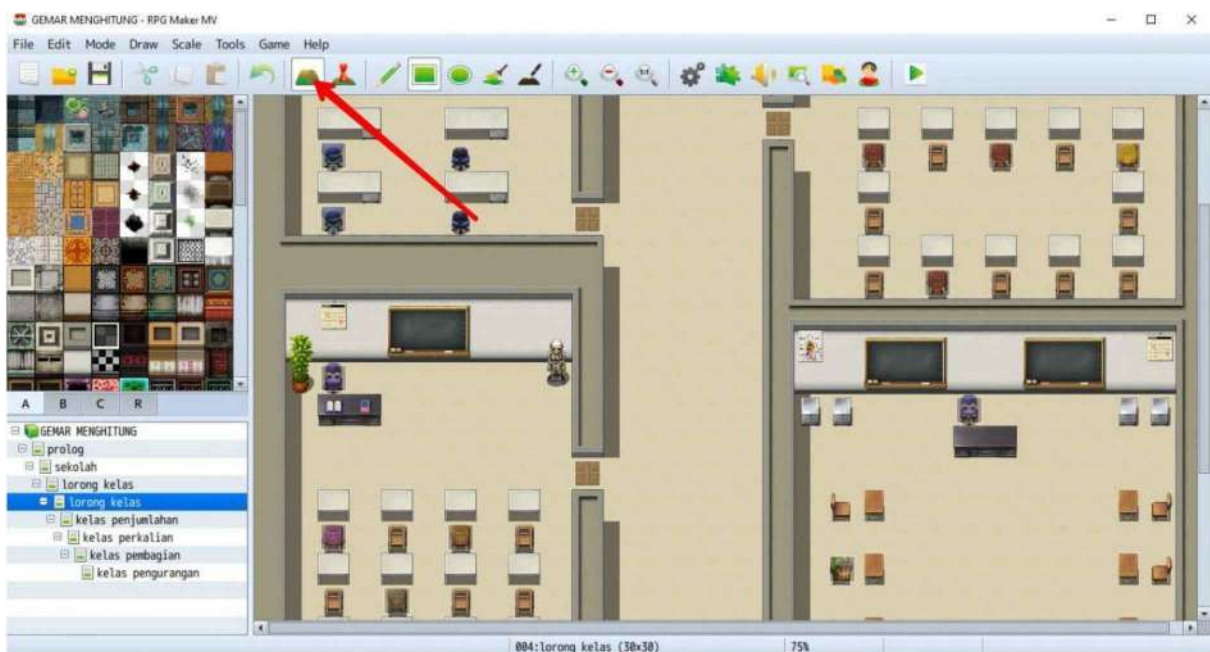
Event adalah tempat untuk membuat peristiwa dalam permainan, seperti melakukan teleportasi ke tempat lain, menjawab soal, dan lain-lain.



Gambar 1.30. *Event*

Map

Map adalah tempat membuat gambaran dalam permainan, seperti membuat bangunan sekolah, kelas, dan lain-lain.



Gambar 1.31. *Map*

Membuat prolog

Prolog adalah tahapan awal dalam membuat sebuah *game*, prolog juga adalah penggambaran awal sebuah *game* akan seperti apa.

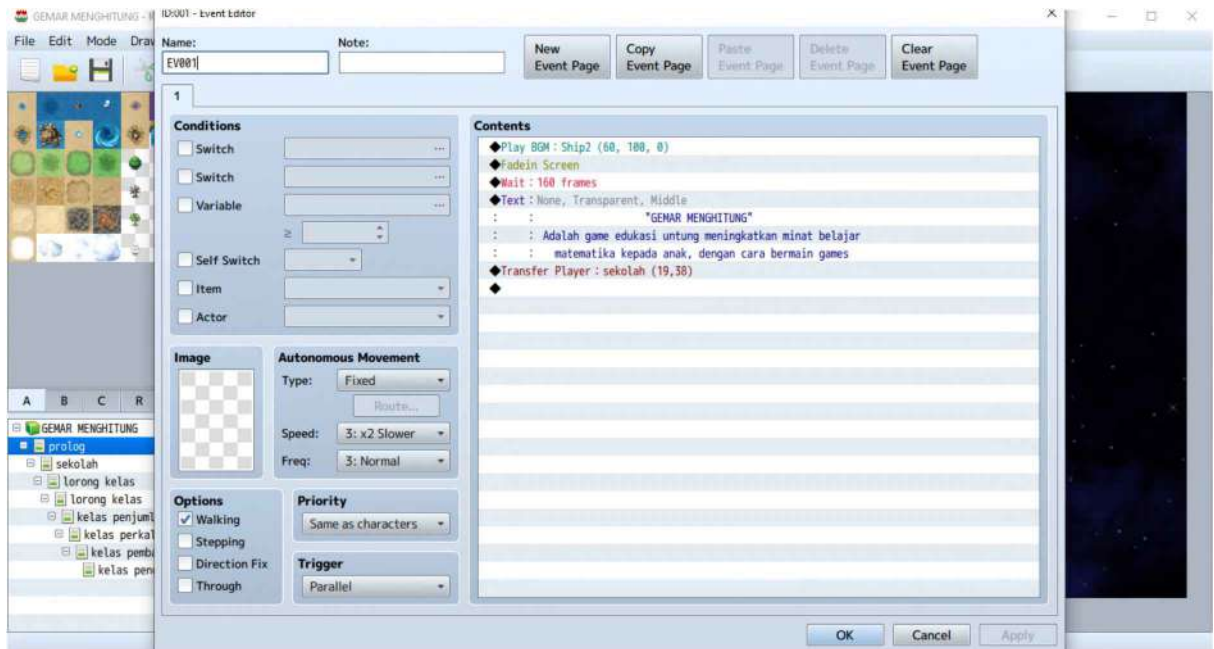
Langkah-langkah membuat prolog:

1. Anda dapat menyunting terlebih dahulu latar ukuran prolognya, dengan cara klik kanan pada bagian prolognya, lalu anda dapat menyuntingnya sendiri, sesuai kesukaan.



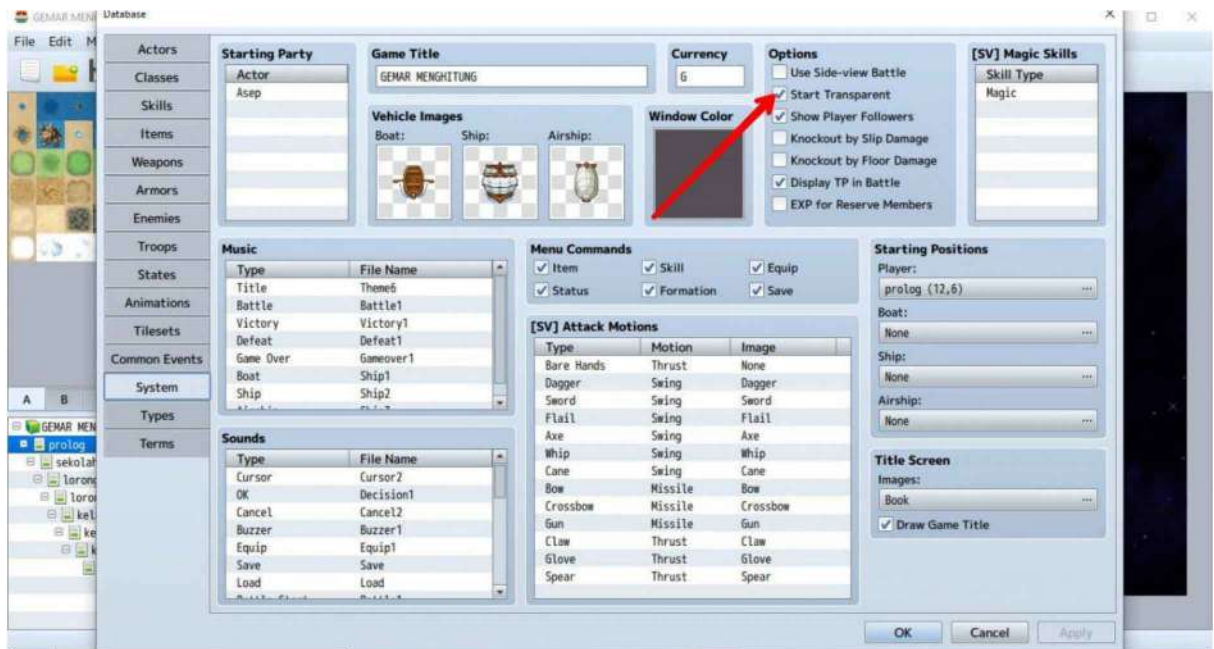
Gambar 1.32. Klik kanan sesuai anak panah untuk meng-*edit*

2. Anda dapat menyunting bagian *event* pada prolog sesuai selera dan kesukaan masing-masing, dan jangan lupa *Priority*nya diubah ke *same as characters*, dan *Trigger*nya di ubah ke *parallel*.



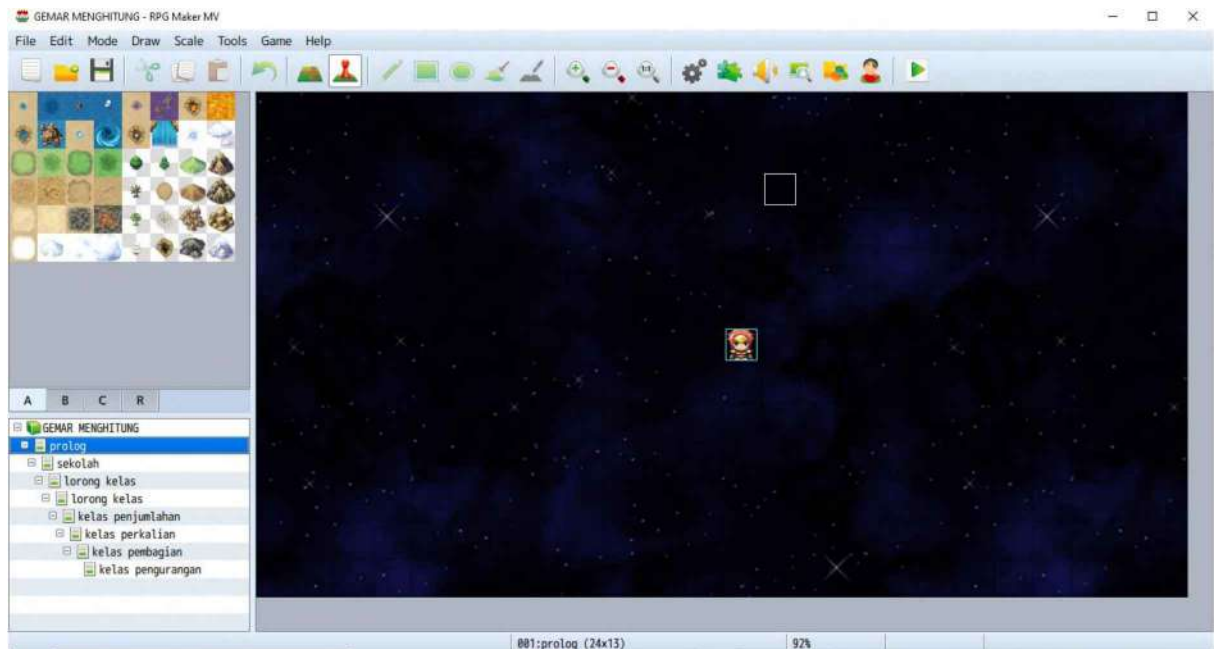
Gambar 1.33. *Contents event page* prolog

3. Agar tidak terjadi *bug*, seperti karakternya muncul pada prolog berlangsung anda dapat mencentang *start transparent* di bagian *Database*, lalu ke *System*, lalu ke *option*.



Gambar 1.34. Mengklik sesuai anak panah

4. Tampilan jika prolog yang *diedit* seperti gambar di atas.



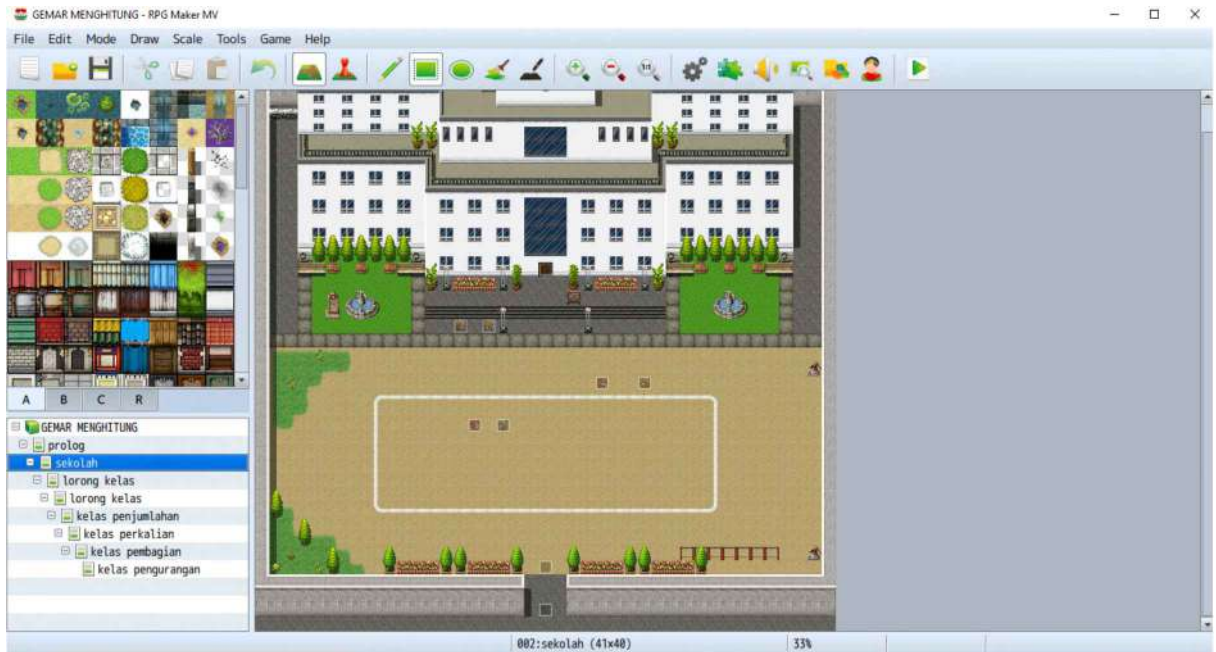
Gambar 1.35. Tampilan prolog

Membuat peta dan NPC

Dalam sebuah *game* pasti ada peta untuk tempat permainan bagi pemainnya. Peta dapat berbentuk apa saja, misalnya bangunan.

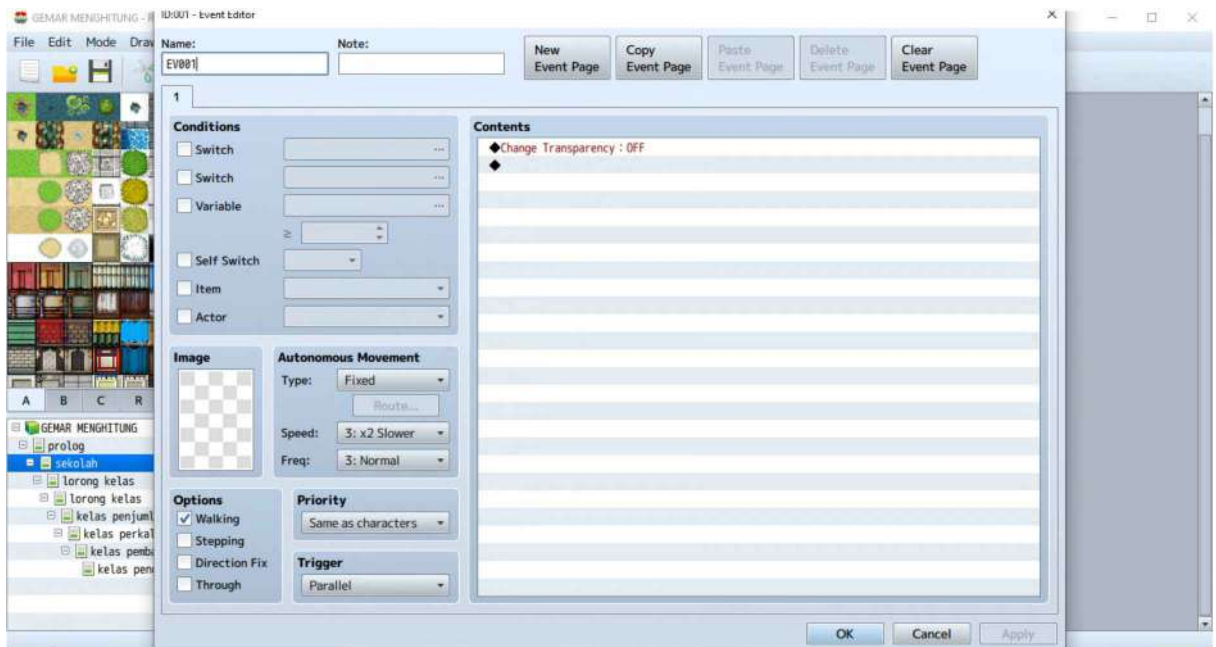
Langkah-langkah membuat peta:

1. Dalam membuat peta (sekolah). Anda dapat memakai bahan-bahan yang ada, dan dapat memakai model-model peta yang sudah tersedia.

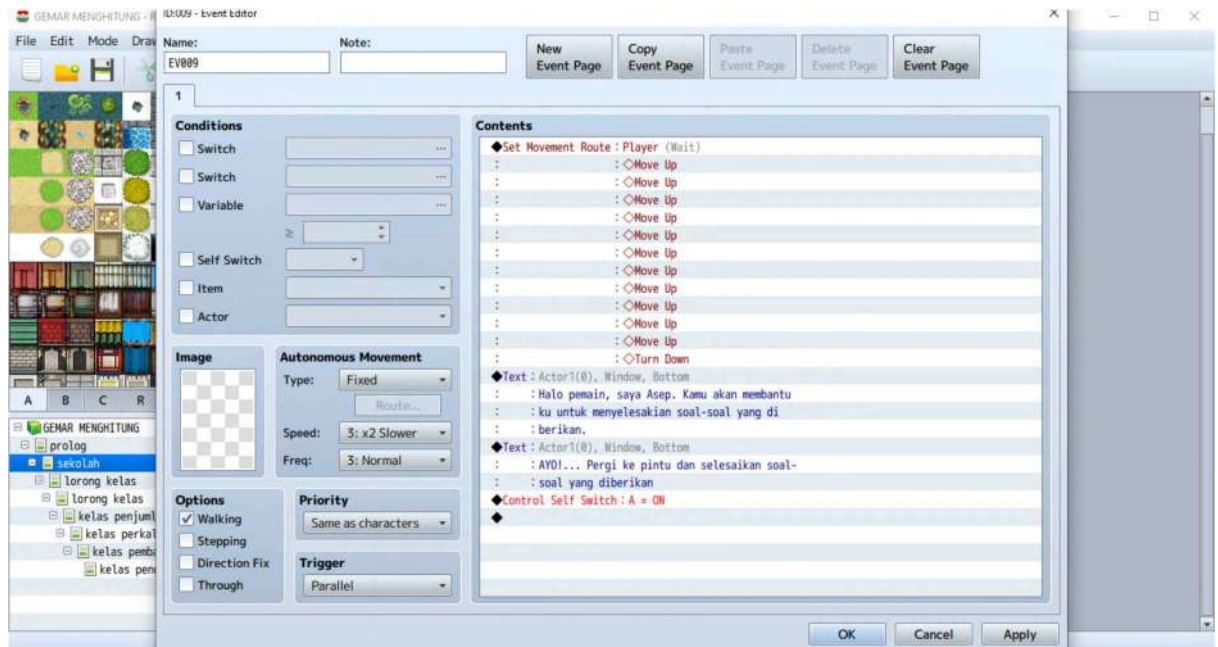


Gambar 1.36. Peta gedung dan halaman sekolah

2. Di dalam peta ini terdapat dua *event*, yaitu *event* memunculkan karakter pemain sehabis prolog, dan *event* selamat datang untuk pemain-pemainnya, dan jangan lupa untuk *Priority* dan *trigger*-nya diganti sesuai dengan *event* yang mau dijalankan seperti apa.



Gambar 1.37. *Contents* untuk memunculkan karakter kembali

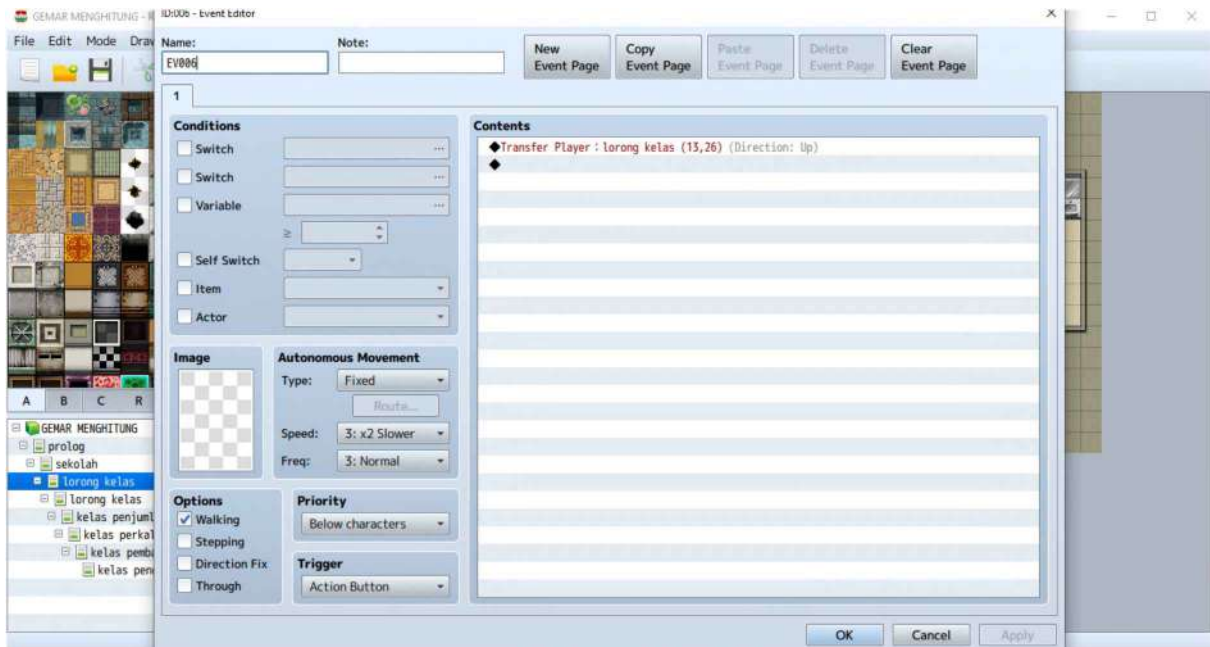


Gambar 1.38. *Contents event page* membuat selamat datang kepada pemain

3. Selanjutnya, kamu dapat membuat Lorong sekolah untuk dapat mencapai ke Lorong kelas. Dan membuat *event* untuk mengteleportasi ke peta selanjutnya.

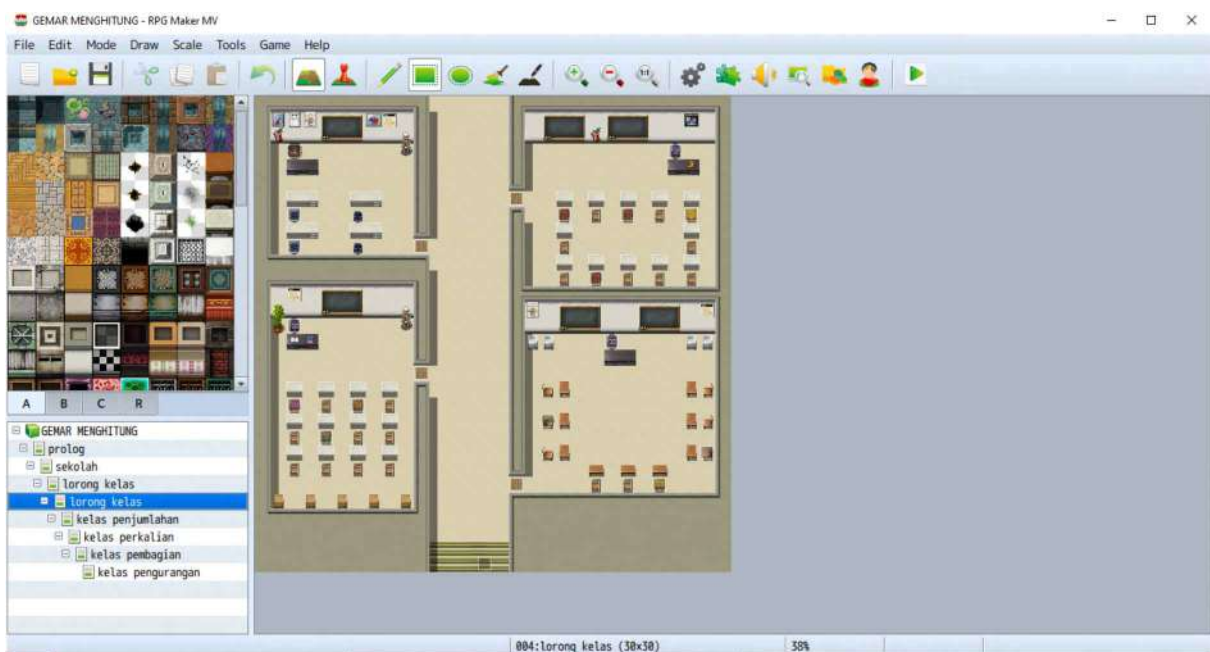


Gambar 1.39. Peta Lorong sekolah



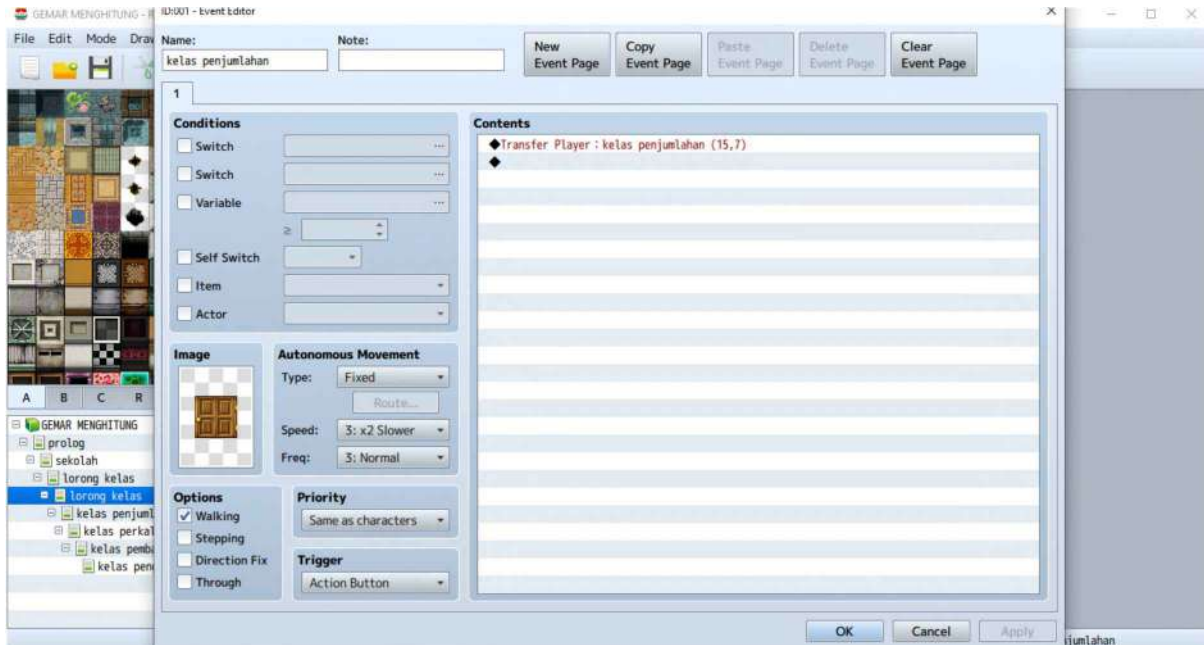
Gambar 1.40. *Contents event page* teleportasi ke peta selanjutnya

4. Membuat peta Lorong kelas dan kelas yang digunakan untuk tempat permainan.

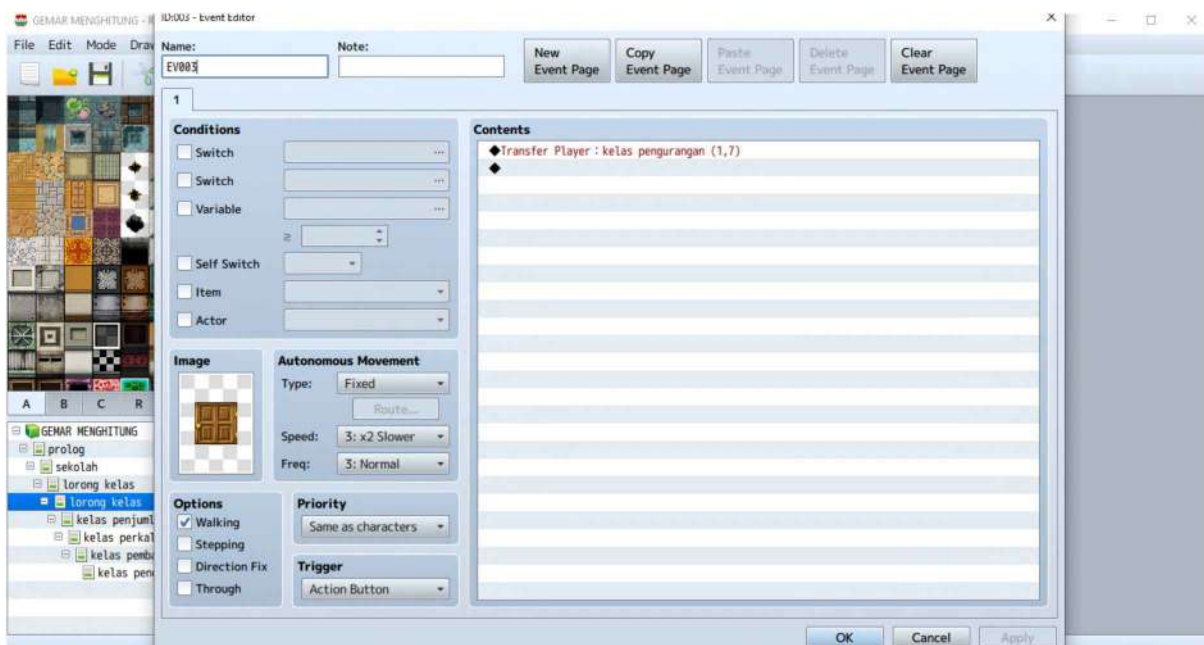


Gambar 1.41. Peta Lorong kelas

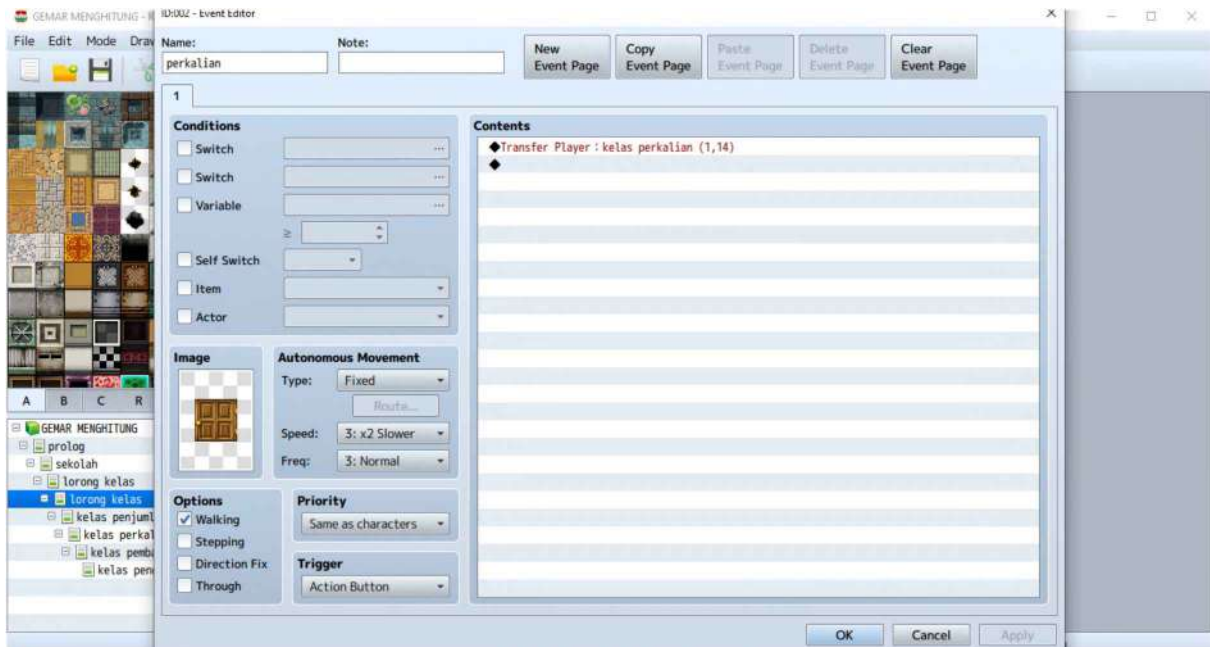
5. Terdapat *event* teleportasi pada pintu, untuk mengteleportasi pemain dari Lorong ke ruangan kelas-kelas yang ada.



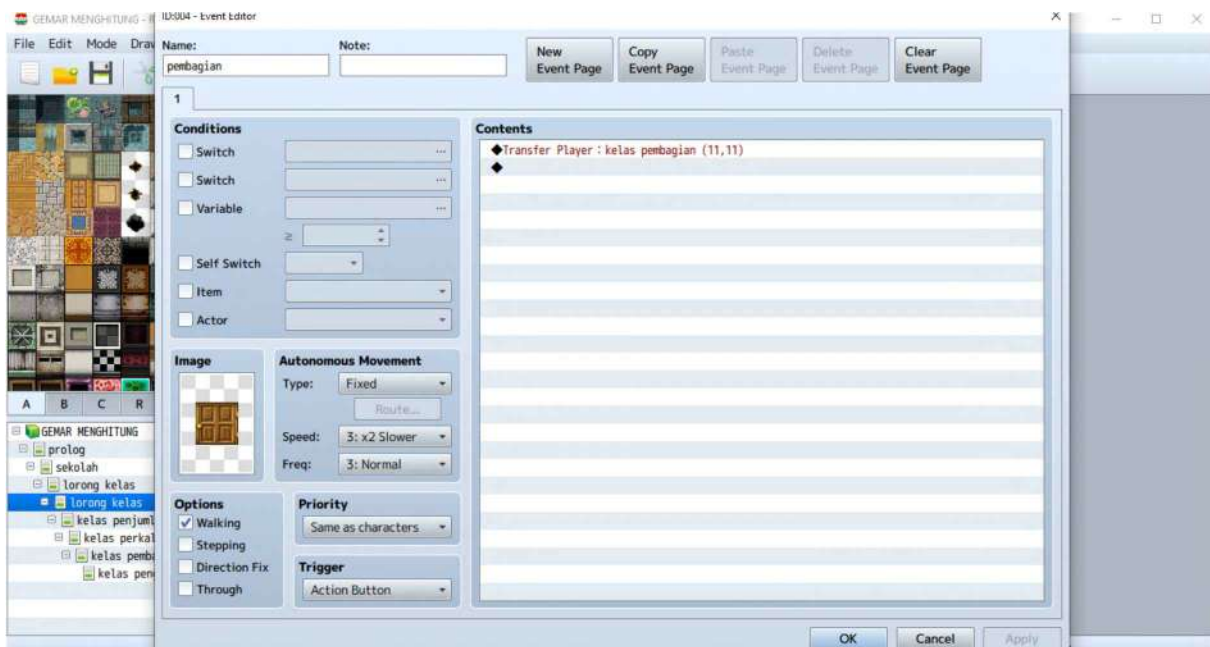
Gambar 1.42. Pintu kelas penjumlahan



Gambar 1.43. Pintu kelas pengurangan



Gambar 1.44. Pintu kelas perkalian



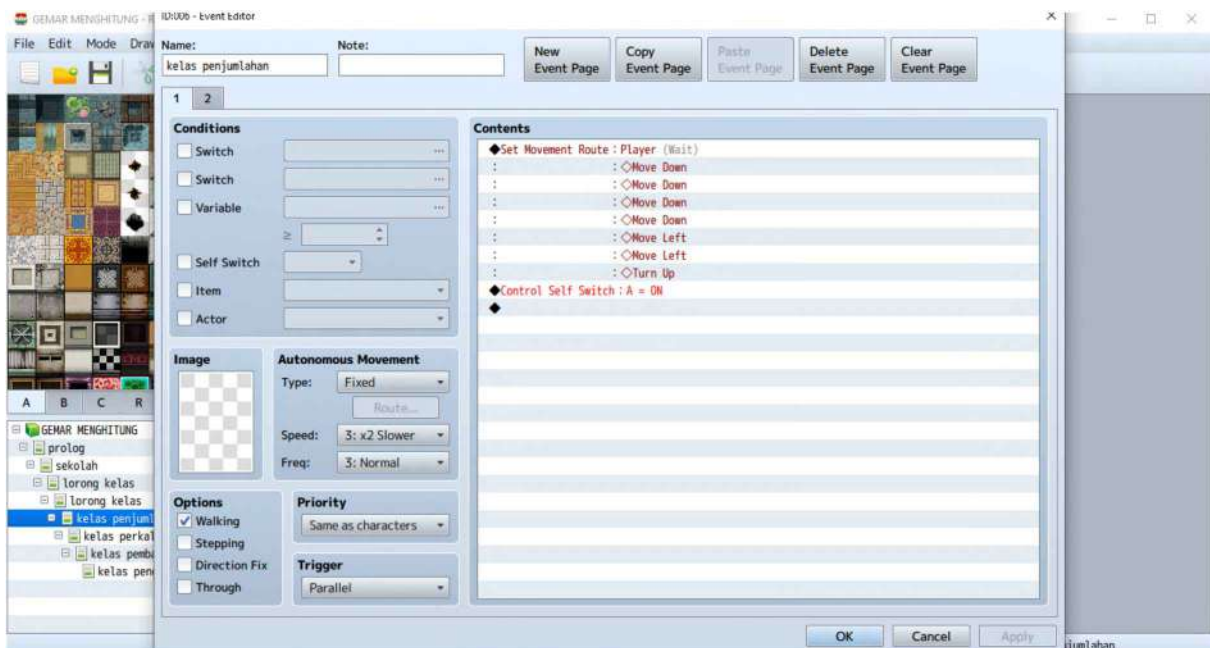
Gambar 1.45. Pintu kelas pembagian

6. Selanjutnya kamu bisa membuat masing-masing ruangan kelas sebagai tempat teleportasi dari pintu yang ada di Lorong tadi.

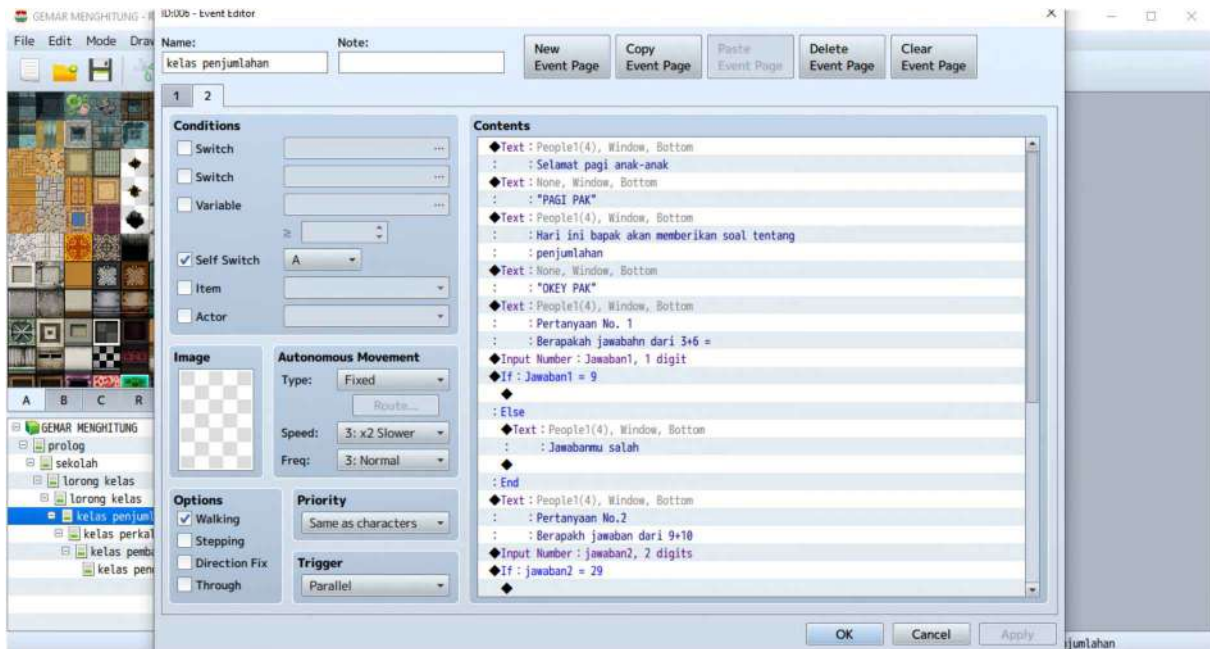


Gambar 1.46. Ruang kelas penjumlahan

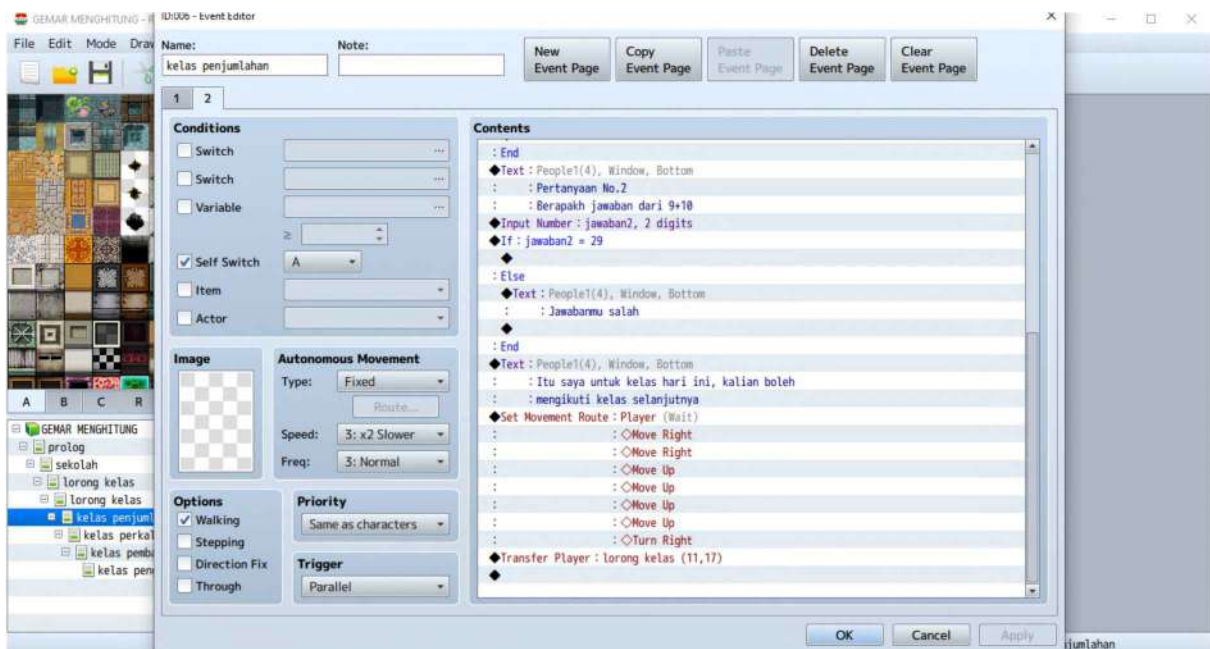
7. Terdapat *event* pada *map* ini, terdapat dua *event page*.



Gambar 1.47. *Contents event page* satu kelas penjumlahan

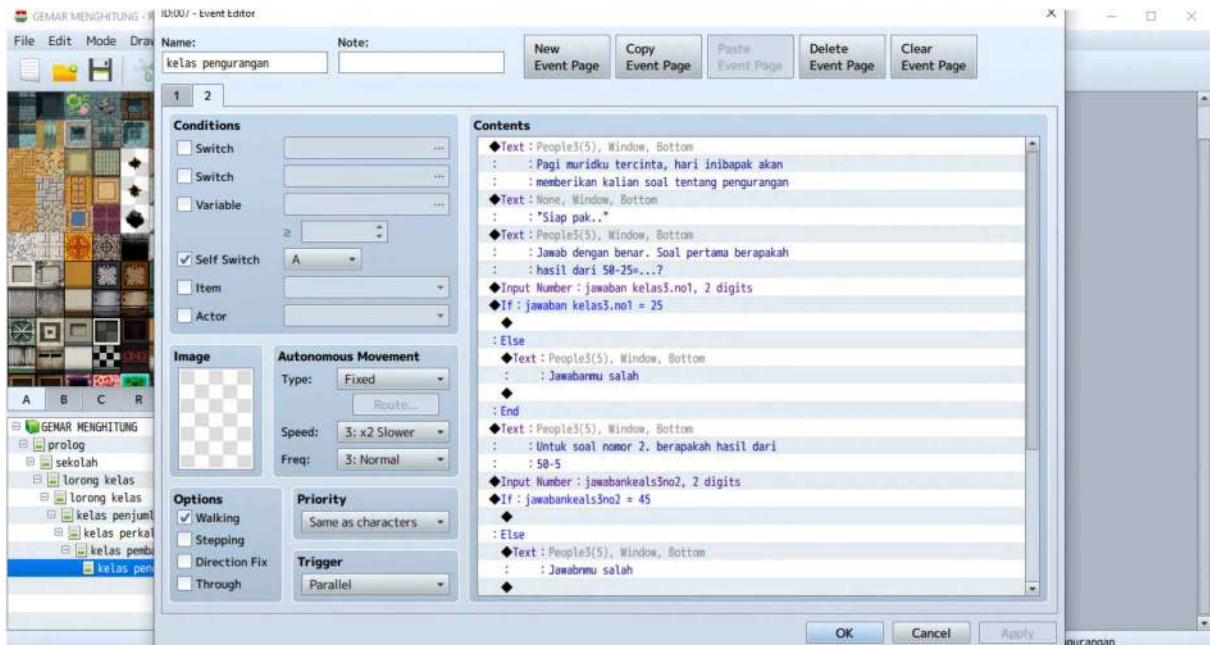


Gambar 1.48. Contents event page dua kelas penjumlahan

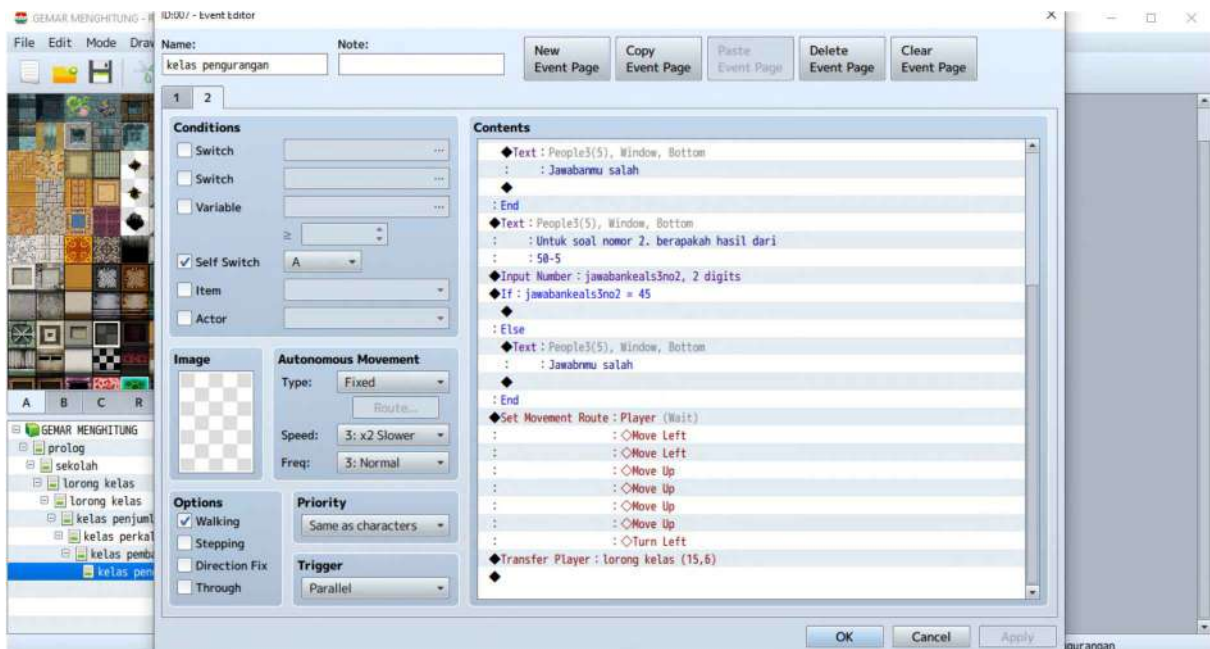


Gambar 1.49. Lanjutan

8. Untuk selanjutnya kamu dapat membuat kelas yang berikutnya.

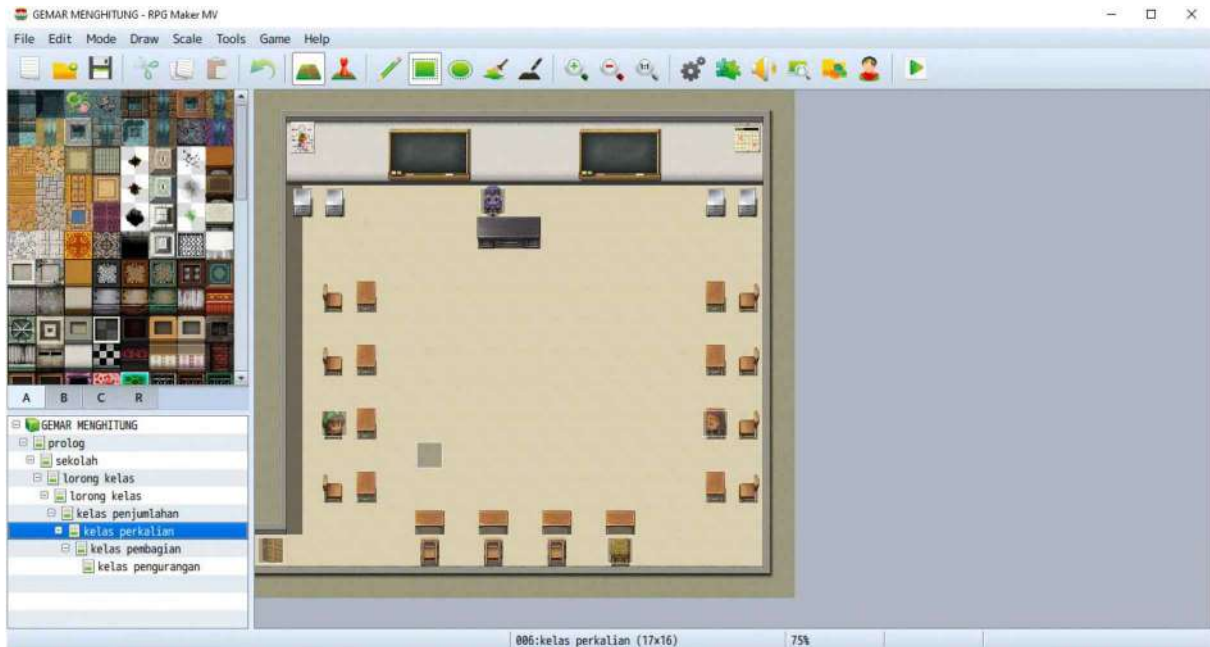


Gambar 1.52. Contents event page dua kelas pengurangan



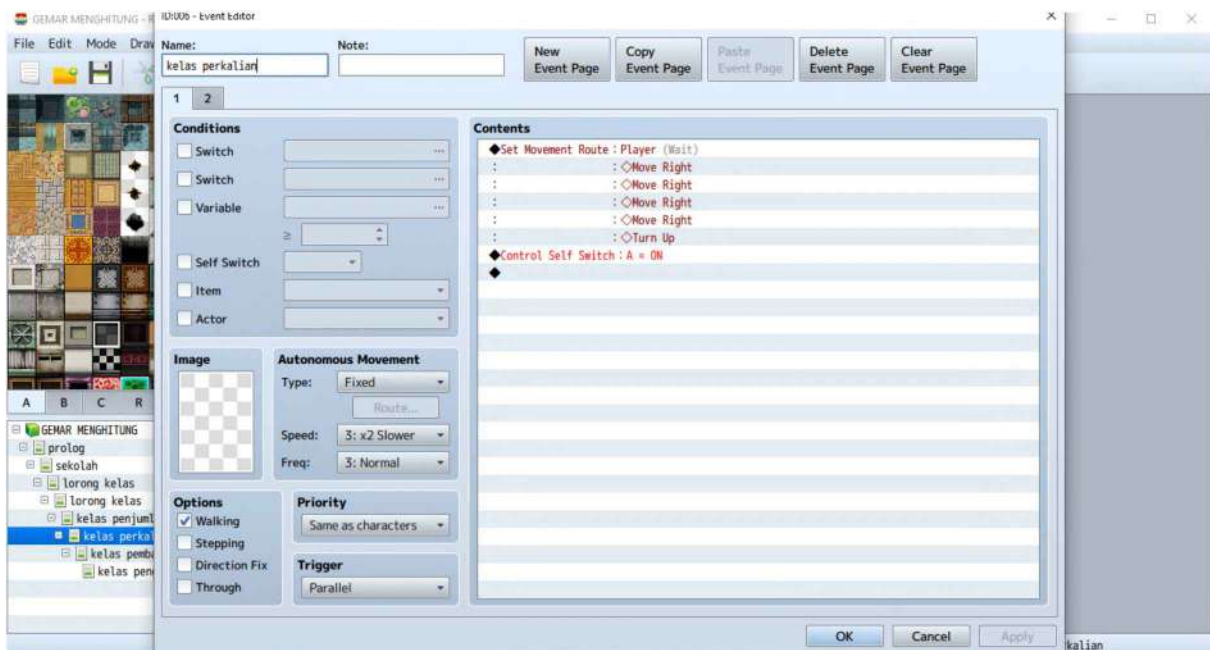
Gambar 1.53. lanjutan

1. Membuat peta ruangan kelas perkalian.

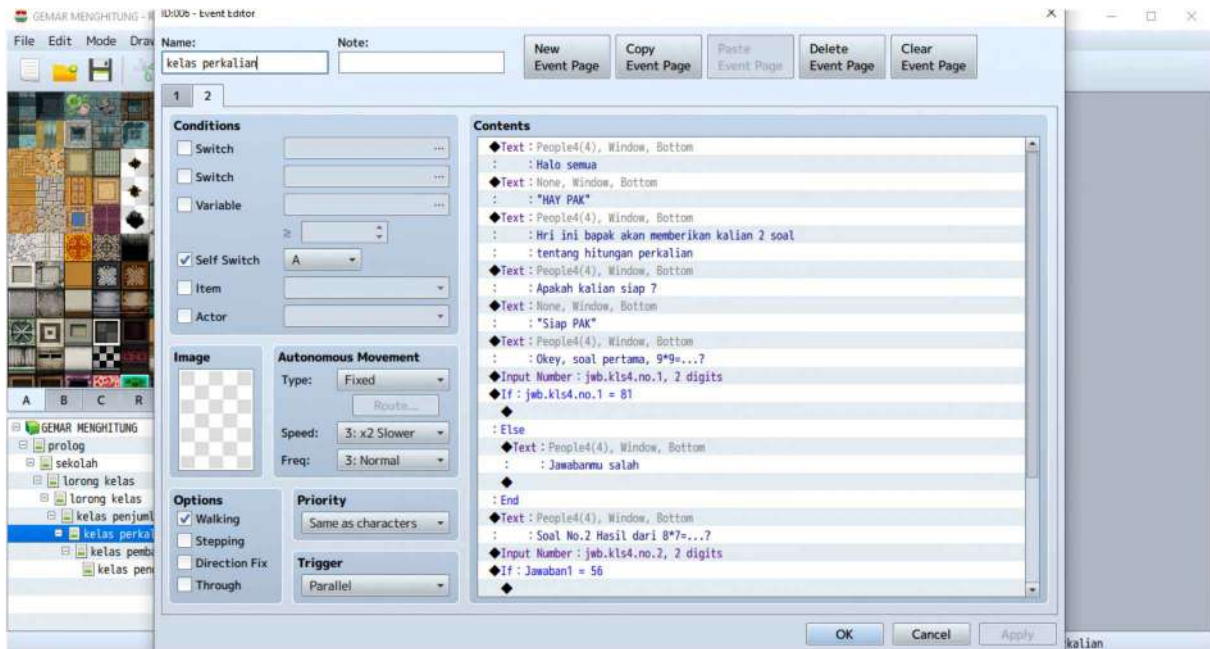


Gambar 1.54. Ruang kelas perkalian

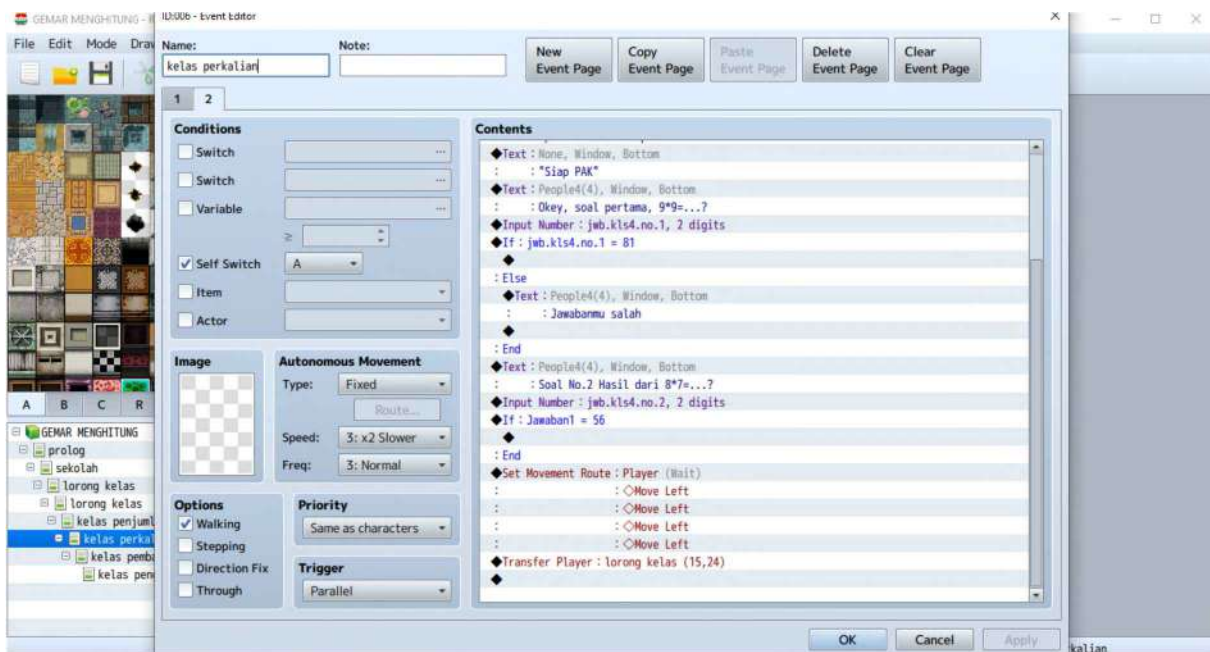
2. Terdapat dua *event page* di peta ini.



Gambar 1.55. *Contents event page* satu kelas perkalian

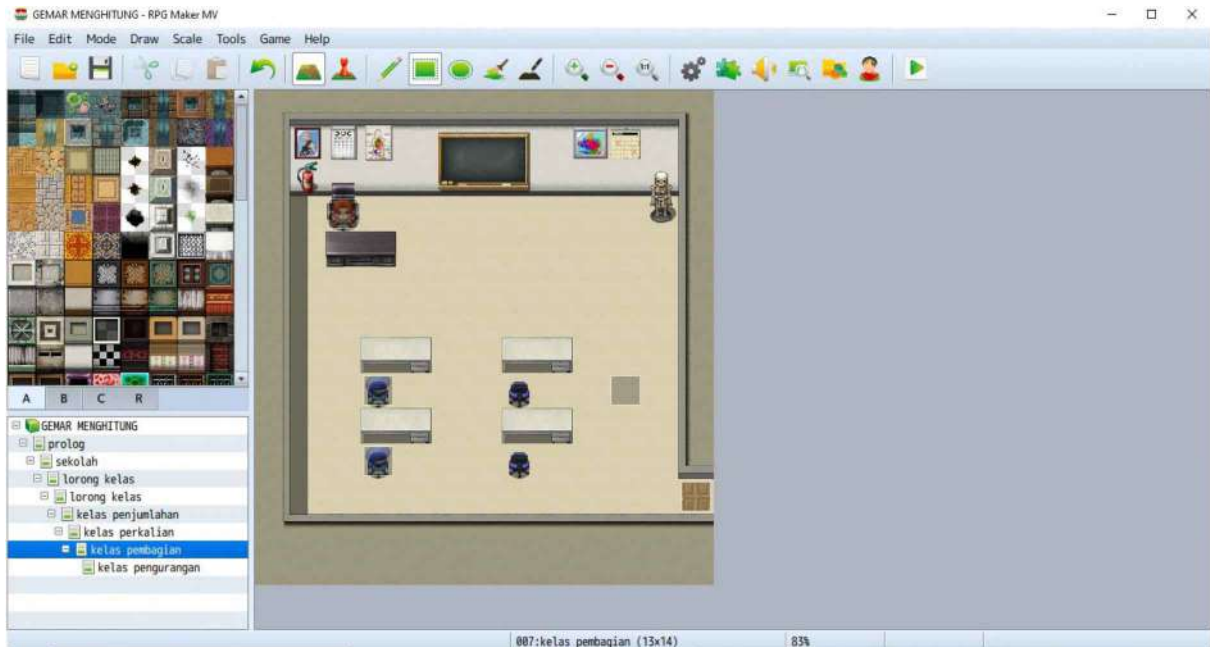


Gambar 1.56. *Contents event page* dua kelas perkalian



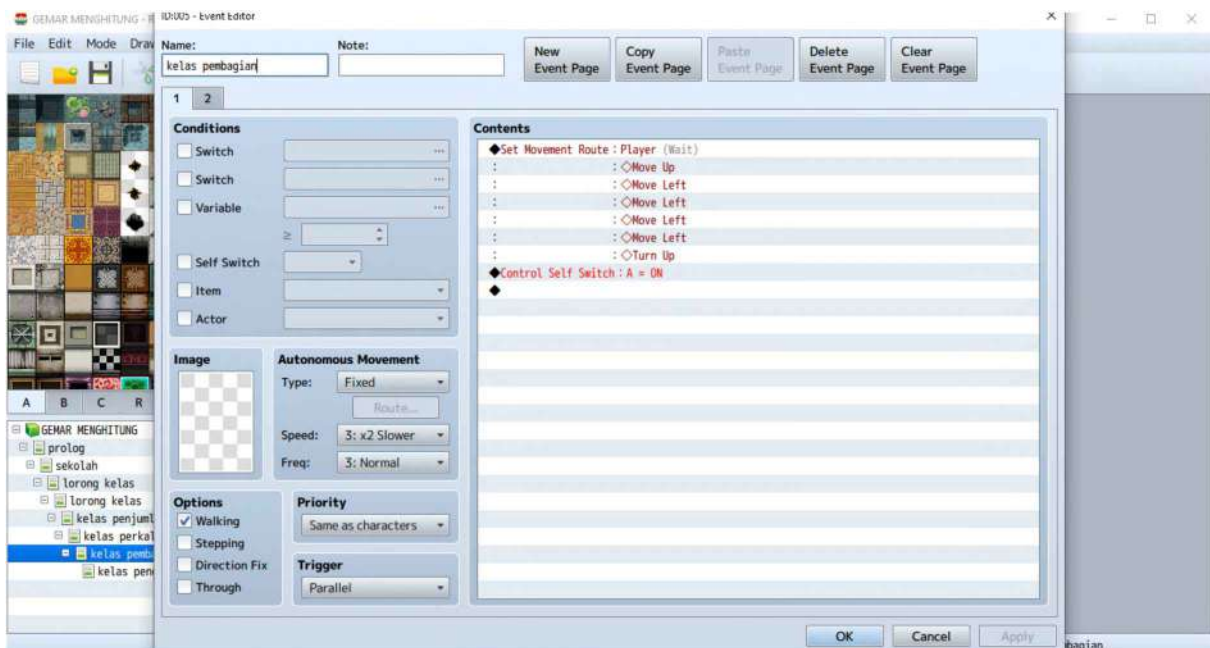
Gambar 1.57. Lanjutan

3. Membuat peta ruangan kelas pembagian.

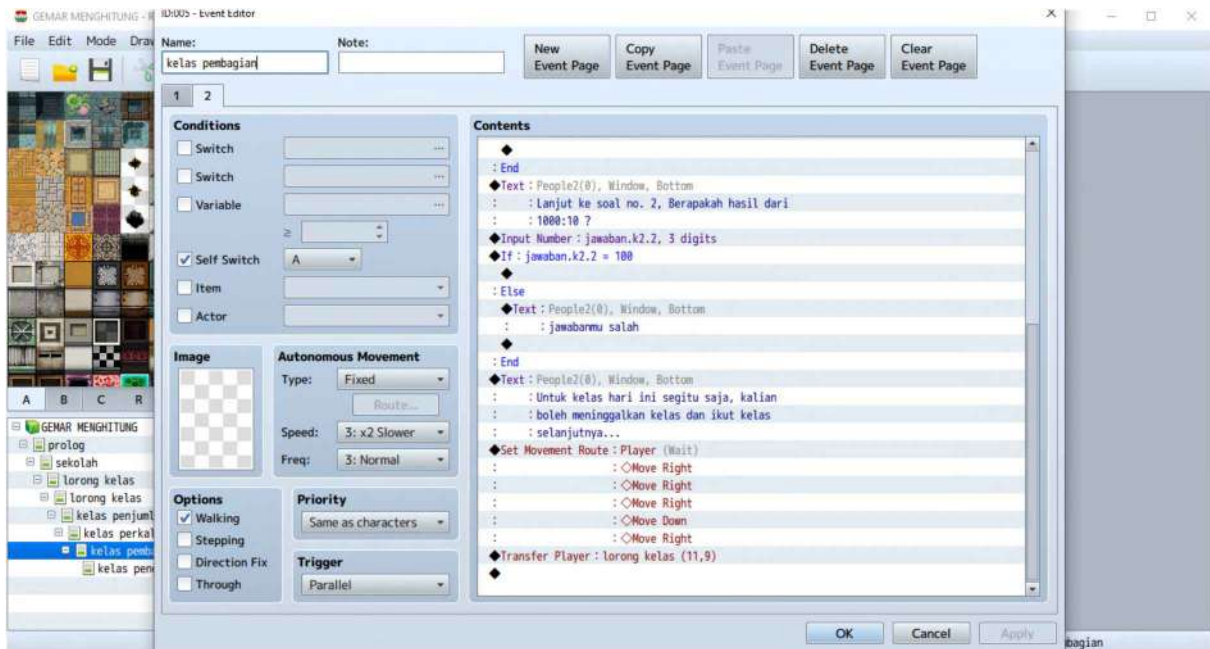


Gambar 1.58. Ruangan kelas pembagian

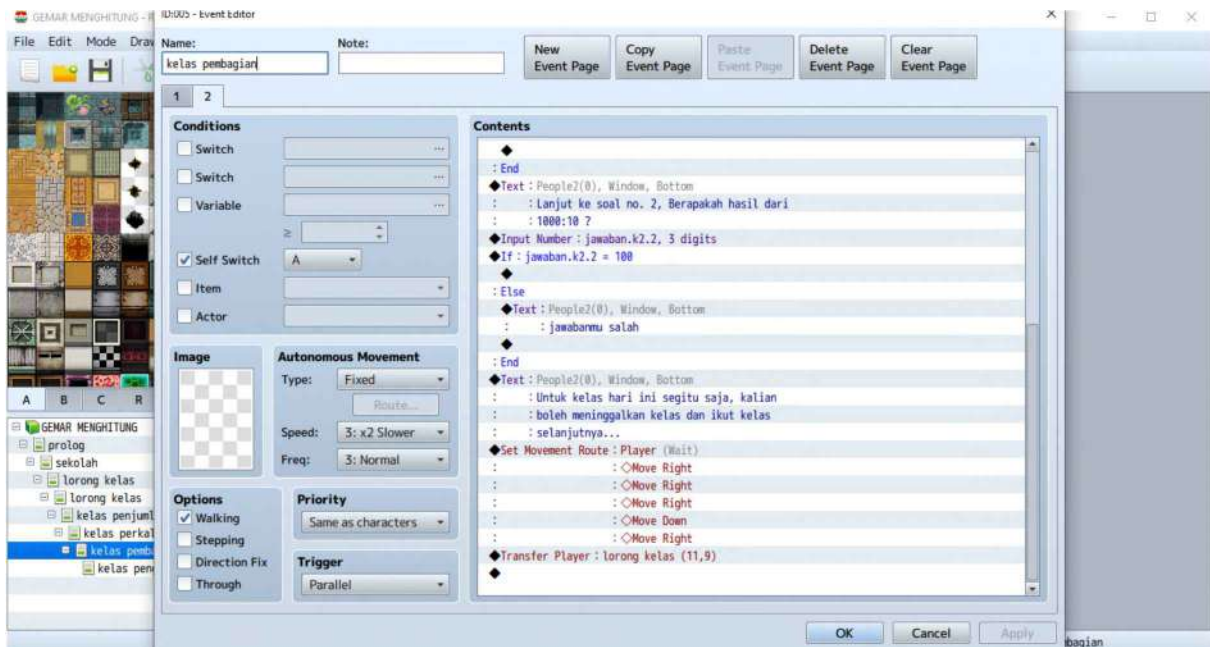
4. Terdapat *event* dalam peta ini.



Gambar 1.59. *Contents event page* satu kelas pembagian

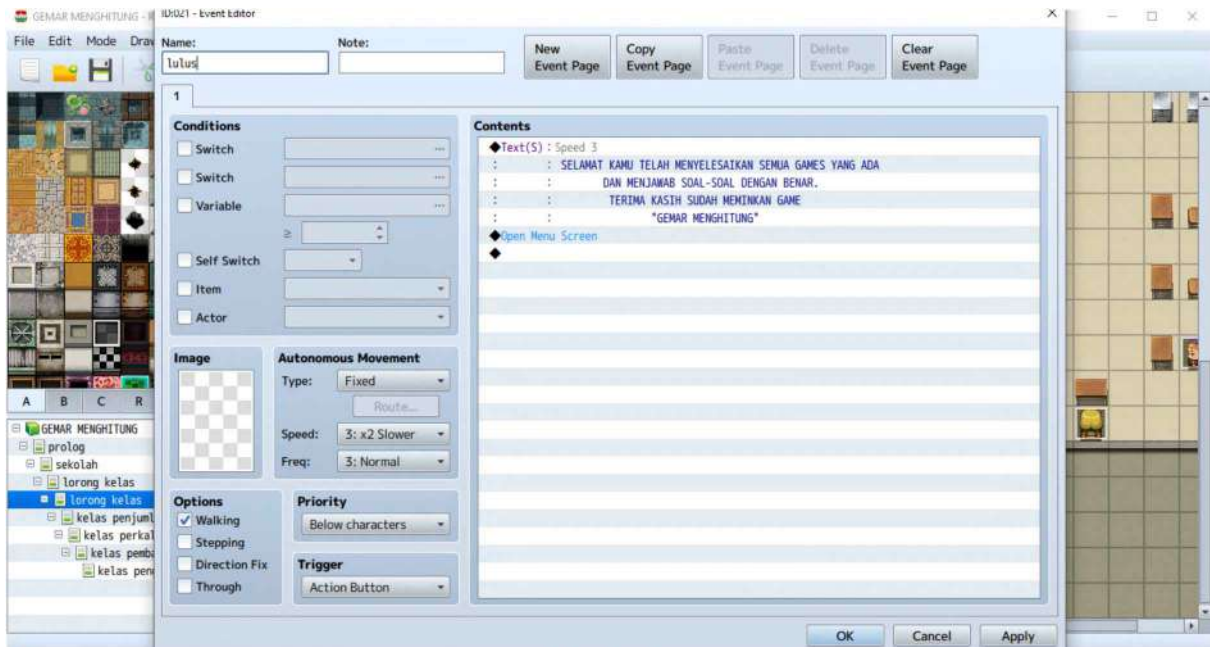


Gambar 1.60. *Contents event page* dua kelas pembagian



Gambar 1.61. Lanjutan

5. Jika dapat menyelesaikan semua soal-soal yang diberikan dengan benar kamu akan dinyatakan lulus/sudah menyelesaikan *game*.



Gambar 1.62. *Contents event page game selesai*

Link download game

Link download 1:

<https://tinyurl.com/464z2k6z>

Link download 2:

<https://drive.google.com/file/d/1Gx5VovG7cLOLoHluMqArhKaAWTVBKjpX/view?usp=sharing>

PILIH JALAN YANG BENAR

Kevin Wijaya

Game adalah sesuatu yang kadang-kadang digunakan untuk bermain bisa berbentuk barang atau sesuatu yang biasa digunakan untuk hiburan terkadang digunakan sebagai alat pendidikan juga. Ada beberapa platform *game* yang saya ketahui yaitu:

- Game Komputer
- Game Mobile
- Game Konsol

Game komputer adalah suatu permainan yang dapat dimainkan di komputer menggunakan *mouse* atau *keyboard* bahkan juga stik untuk *PC (Personal Computer)*. Komputer ini dapat memainkan *game-game* selain dari *PC*, seperti *game-game console* dan *game mobile*, tetapi kelemahan dari *game PC* adalah kontrol tidak nyaman menggunakan *joystick* seperti *game console*.

Game Mobile yaitu permainan yang dapat dimainkan di handphone, kelebihanannya yaitu praktis dan kelemahannya menguras baterai. Ada 5 contoh *game mobile* yang saya ketahui sekarang diantaranya adalah:

- Clash Of Clans
- Pubg Mobile
- Minecraft Pocket Edition
- Mobile Legends

- Modern Warship

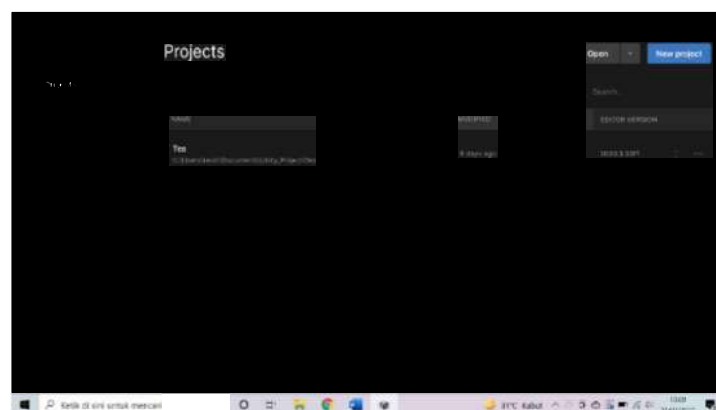
Game Console adalah permainan yang dapat dimainkan dengan menggunakan *controller* atau *joystick*. Kelebihan dari *game console* ini adalah kontrol game lebih optimal, kontrol yang dipegang lebih nyaman karena menggunakan *joystick*. Kelemahannya adalah *gamenya* mahal, dan tidak semua *game PC* dapat dimainkan.

Contoh Game Engine

Untuk membuat *game* diperlukan *game engine* berikut contoh-contohnya

1. Unity

Pembuat Unity adalah Unity Technologies (Unity Wikipedia, n.d.) yang berasal dari Denmark berbasis di San Fransisco, ini adalah *game engine* yang dapat didapatkan secara gratis tetapi juga *powerfull* pada 2005 bulan Juni *unity* diluncurkan ke publik.



Gambar 2.1. Halaman Unity

Game engine ini dapat digunakan untuk membuat game 3D (Tiga Dimensi), 2D (Dua Dimensi), *VR* (Realitas Virtual), *AR* (Realitas Virtual). Bahasa pemrograman *unity* adalah *C#*. Contoh *game mobile* yang dibuat menggunakan *unity* adalah Mobile Legends, Modern Warship, Petualangan Alto, Shadow Fight 3, PAKO, Angry Birds 2 dan lain-lain. (Portdesk.Net, n.d.).

2. Unreal Engine

Epic Games adalah yang membuat *game engine* ini (Unreal Engine Wikipedia, n.d.), pada awalnya *game engine* ini hanya digunakan untuk membuat *game* bertema tembak-menembak orang pertama saja namun akhirnya ada pengembang yang bisa membuat selain tema itu. Dengan kode pemrograman *C++ game* ini mampu membuat game contohnya

- PUBG
- BioShock
- Gears Of War 3
- Borderlands 2
- Batman Arkham City

3. CryEngine

CryEngine sangat cocok jika pengembang ingin membuat *game* bertema perang, dengan dilengkapi keunikan dari CryEngine yaitu pencahayaannya yang cukup bagus untuk dan tentunya memudahkan untuk para *gamers* melihat musuh dan suaranya yang cukup jernih sehingga tidak perlu

memasang speaker besar. *CryEngine* ini menggunakan bahasa pemrograman *LUA* untuk *scripting*. Contoh-contoh *game* yang dibuat dari *CryEngine* adalah (Gamebrott.Com, n.d.)

- Far Cry
- Cabal 2
- Sniper Ghost Warrior 2
- Crysis Series

4. Rpg Maker

RPG Maker MV (RPG Maker Wikipedia, n.d.) adalah *game engine* sederhana digunakan untuk membuat *role play gaming*. *Game engine* ini sangat mudah dan cocok untuk *developer* pemula untuk membuat *game RPG*. Contoh *game* yang dibuat dengan RPG Maker ini yang saya ketahui adalah (Getective.Com, n.d.)

- Ib
- The Witch's House
- To The Moon

LATAR BELAKANG GAME

Game puzzle adalah sebuah permainan yang berisi teka-teki untuk memecahkan sesuatu. *Game* yang akan saya buat ber-*genre puzzle*, *game puzzle* adalah sebuah permainan yang berisi teka-teki untuk memecahkan sesuatu. Alasan saya memilih *genre puzzle* karena selain melatih kesabaran, juga meningkatkan daya ingat. Saya menggunakan RPG Maker MV

untuk membuat *game* ini, karena tampilannya yang tidak rumit dan cocok untuk membuat *game* seperti tebak-tebakan, kuis, dan lain-lain. Model *game*-nya yaitu pemain akan menjalankan misinya dengan menjawab berbagai kuis untuk bisa mencapai objektif. Jika pemain benar menjawab kuis akan langsung pindah tempat atau terbuka jalan baru, tetapi jika salah menjawab akan langsung mendapat serangan bermacam-macam seperti misalnya terbunuh oleh *monster-monster* atau tempatnya menjadi berbahaya atau akan terbunuh dengan sendirinya dan akan otomatis gagal menjalankan misi dan pemain harus mengulang dari awal. *Game* ini juga dapat membantu melatih kemampuan ingatan, karena akan terdapat kuis dimana pemain diminta bertanya kepada *NPC* lain untuk menuju jalan yang benar kemudian diingat-ingat dan diberitahu kepada *NPC* satunya lagi yang memberi kuis tersebut.

Jalan Cerita

Jalan cerita utama dari *game* “Pilih Jalan Yang Benar” adalah pada awalnya ada seorang remaja berusia 20 tahun Ia bernama Simo, ia tinggal di rumah yang sederhana bersama adiknya bernama Somi, Somi suka bermain dan berpetualang dan Simo suka dengan tantangan dan berpetualang.

Pada suatu hari adik Simo bernama Somi ingin pergi jalan-jalan ke luar kota dan berjanji kepada Simo kembali ke rumah besok pagi, Simo mengizinkannya dengan syarat harus berhati-hati karena kota itu sangat misterius. Adik Simo pun pergi dengan

berpamit ke Simo. Sedangkan Simo menunggu di rumah, suatu ketika lima hari sudah berlalu Simi tidak kunjung kembali.

Simo sangat curiga dan takut adiknya tersesat, ia memutuskan untuk pergi ke kota itu yang bernama kota Kerson, kota dimana adiknya pergi ke sana. Pagi pun tiba kini Simo siap untuk mengunjungi kota Kerson untuk mengambil adiknya yang hilang 5 hari tersebut. Simo tidak tahu sama sekali dengan jalan yang ada pada kota Kerson itu tetapi Simo tidak menyerah dia akan tetap berjalan menuju kota Kerson itu untuk menjemput adiknya. Simo disini harus berhati-hati karena jika sampai salah jalan akan berakibat *fatal* karena kota itu sangat misterius. Ada 2 *ending* dalam *game* ini yaitu

1. *Good Ending*, setelah Simo melalui blok-blok di daerah itu Simo akhirnya menemukan Simi dalam keadaan masih hidup di dalam sebuah *bunker*, dan Simo pun mengantar Simi menuju kembali ke rumah mereka dengan selamat dan hidup bahagia.
2. *Bad Ending*, Simo terbunuh oleh *monster-monster* di daerah tersebut dan Simi tidak ditemukan alias menghilang selamanya.

TANTANGAN GAME

Di *game* ini nantinya pemain akan ditaruh di sebuah kawasan atau kota yang terdiri dari 4 blok. Pemain akan melalui blok-blok itu secara bertahap, setiap blok memiliki tingkat kesusahannya sendiri dan setiap blok terdapat berbagai *NPC* sebagai bagian dari kuis setiap blok juga terdapat penjaga yang

akan membantu pemain melewatinya. Pertama pemain mencari dahulu penjaga blok nya, setelah bertemu dengan penjaga bloknya pemain akan diberikan kuis.

Pemain diharuskan menjawab kuis dengan benar, misalnya pemain berada di blok 1 dan bertemu dengan penjaga blok 1 itu yang memberikan petunjuk cara melewati blok tersebut, misalnya blok 1, petunjuknya adalah harus menemui seorang yang memegang kunci dengan ciri-ciri yang telah diberikan oleh *NPC* penjaga awal yang bertemu dengan pemain, setelah bertemu dengan seorang pemegang kunci tersebut yang sesuai dengan ciri-ciri, pemain harus menemui kembali *NPC* penjaga yang awal tadi untuk membuka pintu jalan menuju blok berikutnya. Jika pemain salah dalam memilih orang yang disebutkan ciri-cirinya oleh *NPC* penjaga maka akan terjadi sesuatu terhadap pemain misal dikejar oleh *monster* tidak terlihat yang akan aktif dan terlihat jika pemain salah dalam menjawab atau akan terbuka pintu menuju jalan yang tidak seharusnya atau jalan menuju jebakan dan akibatnya yaitu pemain gagal dalam menjalankan misi dan harus mengulang dari awal blok.



Gambar 2.2. Peta *Game*

KARAKTER PEMAIN

Pemain utama dalam *game* ini bernama Simo, Ia memiliki sifat yang bertanggung jawab dan berani. Simo berasal dari keluarga yang sederhana dan kini Ia tinggal bersama adiknya yang bernama Somi, mereka hanya tinggal berdua karena kedua orang tuanya menghilang secara misterius setelah mengunjungi kota Kerson yang misterius itu. Simo dan Somi tinggal di rumah yang cukup sederhana di dekat kota Kerson. Somi memiliki sifat yang suka berpetualang dan terkadang pelupa, Simo berusia sekitar 20 tahun sedangkan Somi berusia sekitar 15 tahun.

Tabel 2.3. Tabel Karakter

Nama Karakter	Sifat Karakter	Latar Belakang Karakter
Simo (Pemain) 	Berani dan Bertanggung Jawab.	Kakak kandung Somi.
Somi (NPC) 	Suka berpetualang dan terkadang pelupa.	Adik kandung Simo.

TAHAPAN PEMBUATAN GAME

Dalam pembuatan *game* ini ada beberapa tahapan mulai dari pembuatan peta, *scripting*, pembuatan *level*, hingga mengekspor *game* supaya bisa dimainkan oleh orang lain.

Pembuatan Peta

Di dalam RPG Maker MV terdapat dua *mode* yaitu *Map* dan *Event*, disini saya menggunakan *mode Map* karena saya membuat atau mendesain peta *game*-nya serta menggunakan *database tileset* untuk mengganti *tile* yang diinginkan. Ada 7 tahap Peta yang saya gambar berikut tahapannya:

Tahap 1:

Pembuatan Peta Blok 1



Gambar 2.4. Peta Blok 1

Menggunakan *tileset 0001 Overworld*, membuat pulau terlebih dahulu yang berwarna hijau dengan menggunakan *mode map*, membuat jalan yang berwarna putih memberi pintu di ujung jalan untuk menuju blok 2, membuat pintu menggunakan *mode event*.

Tahap 2:

Pembuatan Peta Blok 2

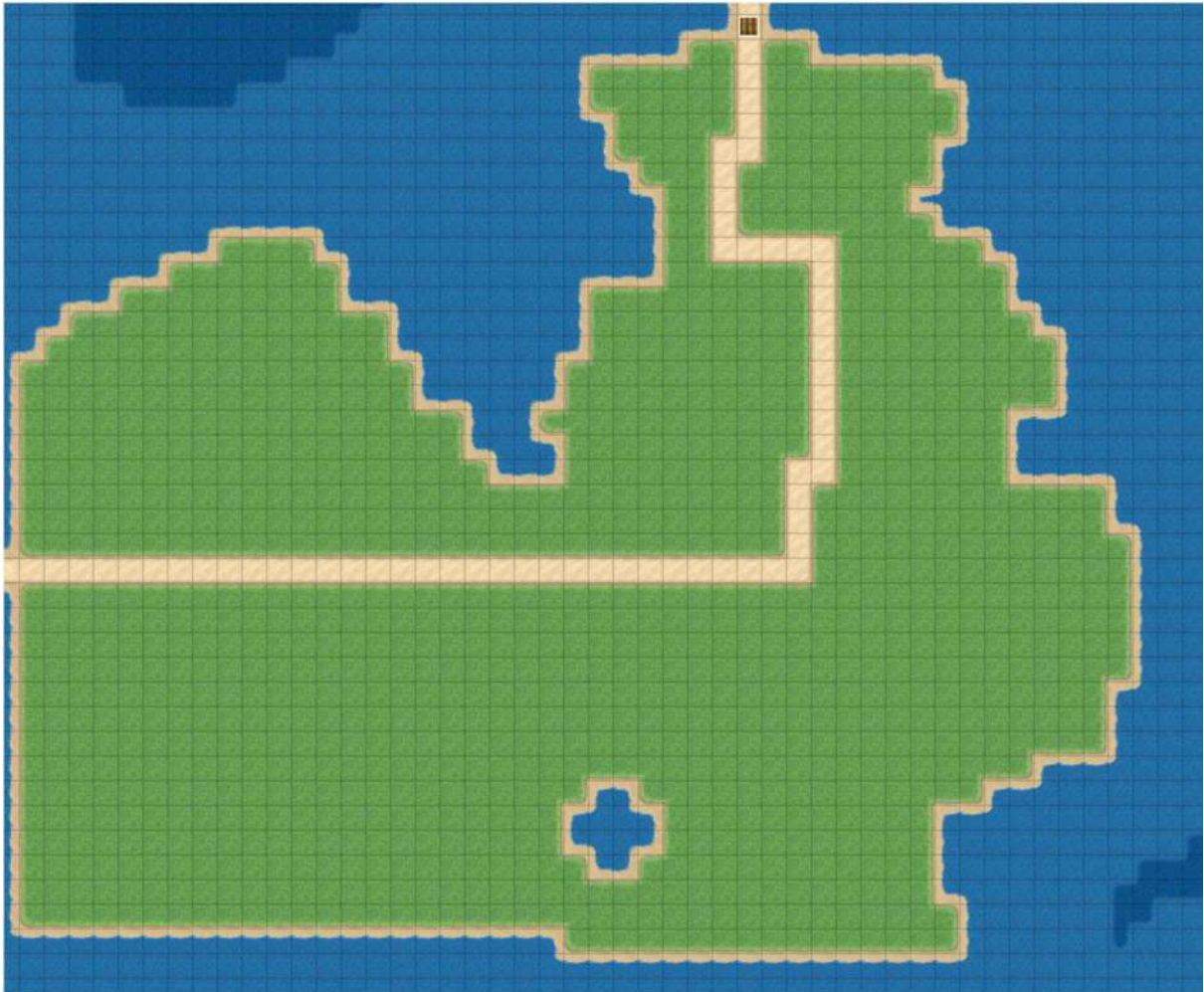


Gambar 2.5. Peta Blok 2

Menggunakan *tileset 0001 Overworld*, membuat pulau terlebih dahulu yang berwarna hijau kemudian membuat jalan yang berwarna putih dan yang terakhir memberi pintu menuju blok 3 di ujung jalan dengan menggunakan *mode event*.

Tahap 3:

Pembuatan Peta Blok 3

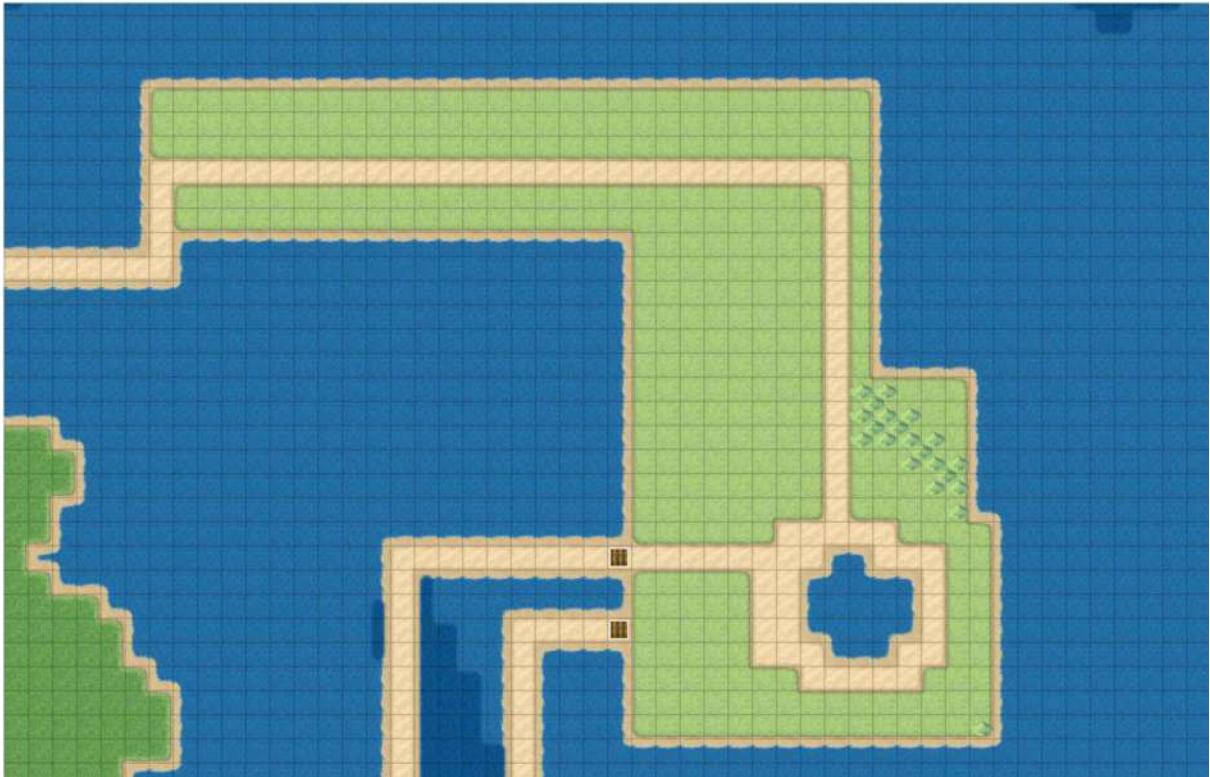


Gambar 2.6. Peta Blok 3

Menggunakan *tileset 0001 Overworld*, terlebih dahulu membuat pulaunya yang berwarna hijau kemudian membuat jalan yang berwarna putih dan yang terakhir memberi pintu menuju blok 4 ditaruh di ujung jalan, membuat pintu menggunakan *mode event*.

Tahap 4:

Pembuatan Peta Blok 4

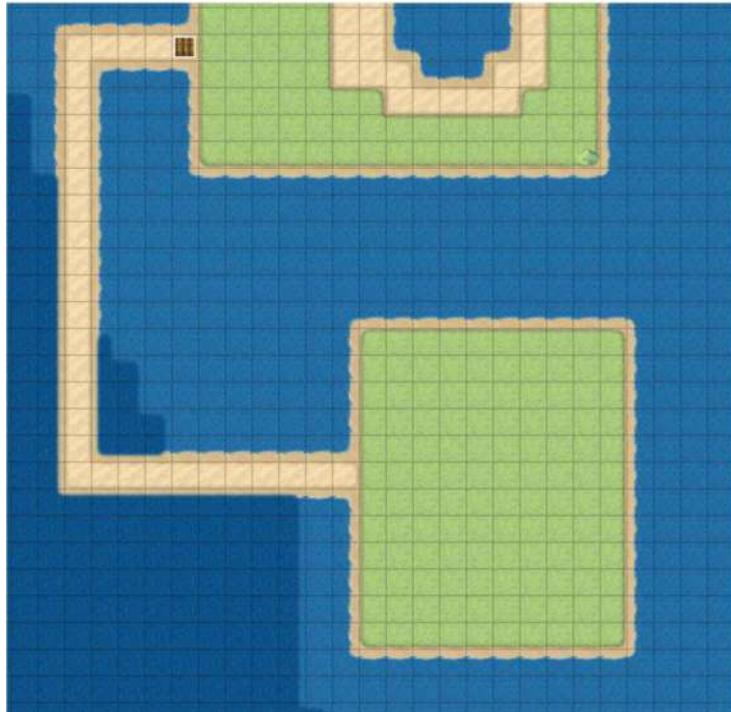


Gambar 2.7. Peta Blok 4

Menggunakan *tileset 0001 Overworld*, membuat pulau yang berwarna hijau dahulu kemudian membuat jalan yang atas jalan yang benar yaitu menuju blok selanjutnya kemudian jalan yang bawah yaitu jalan yang salah, menuju jebakan. Dan yang terakhir memberi pintu diujung jalan.

Tahap 5:

Pembuatan Peta Jebakan Di Blok 4

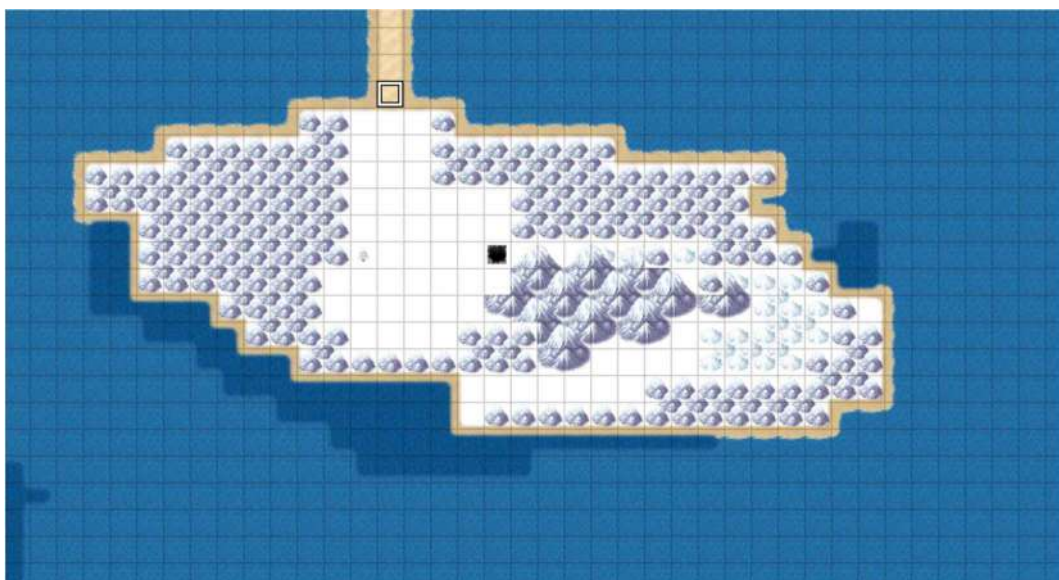


Gambar 2.8. Peta Jebakan Di Blok 4

Membuat pulau yang berwarna hijau dahulu kemudian menyambungkan dengan blok 4 menggunakan jalan yang berwarna putih.

Tahap 6:

Pembuatan Peta Blok 6

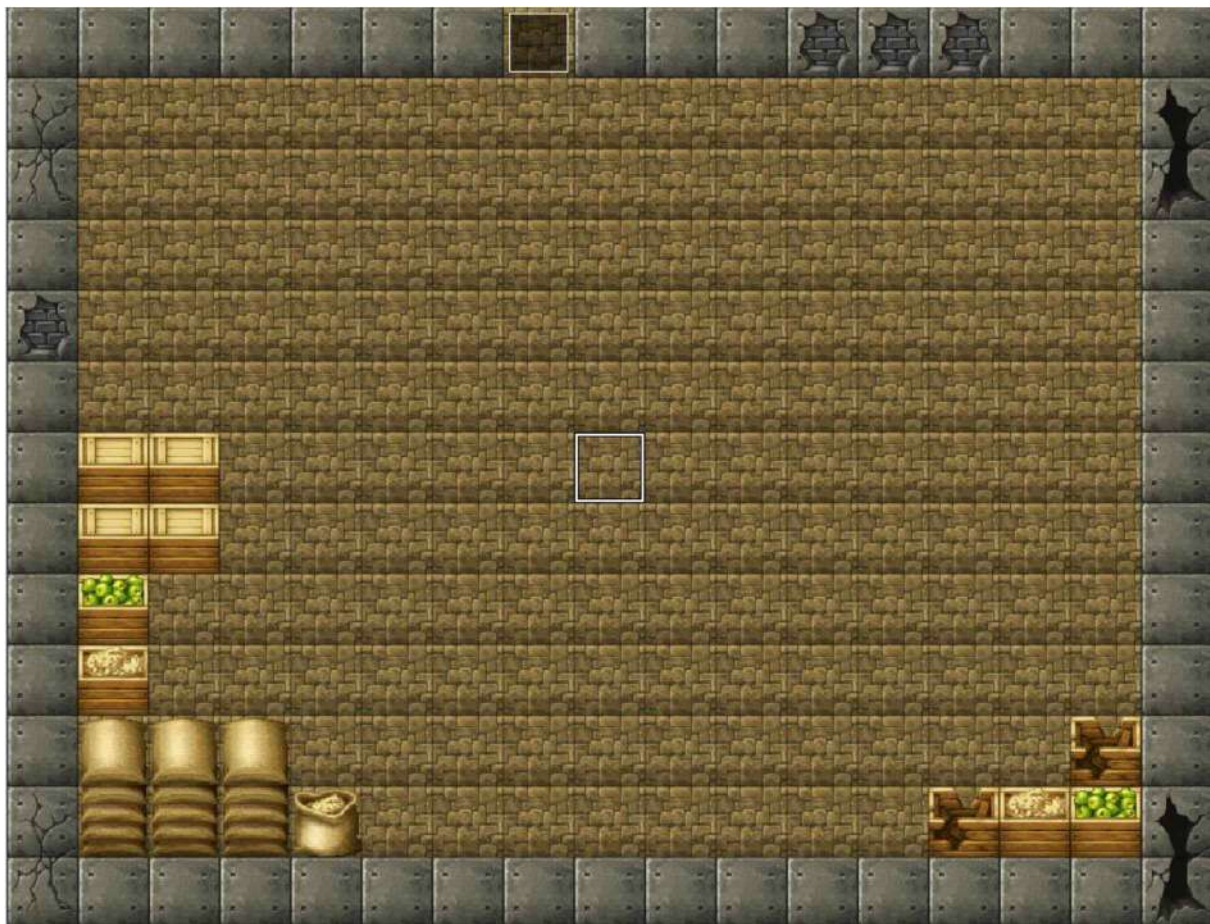


Gambar 2.9. Peta Blok 5 Salju

Menggunakan *tileset 0001 Overworld*, membuat pulau salju nya dahulu yang berwarna hijau, kemudian memberi 1 lubang dengan menggunakan *mode event*, lubang ini digunakan untuk pemain masuk ke *bunker* dimana Somi berada nantinya.

Tahap 7:

Pembuatan Peta *Bunker*



Gambar 2.10. Peta *Bunker* Somi

Menggunakan *tileset 0003 Inside*, pertama membuat dahulu tembok yang di pinggir berwarna abu-abu dan lantainya yang

berwarna coklat dan terakhir hiasan sumber makanan di dalam *bunker* tersebut.

Pembuatan Level

Level dari setiap blok tentu berbeda-beda, ada 5 tahap *level* yang saya buat.

1. Mencari *NPC* yang disebutkan (Blok 1)

Nantinya pemain akan diberi petunjuk oleh penjaga blok 1 berupa ciri-ciri dari *NPC* yang harus ditemukan.



Gambar 2.11. *Level* Blok 1

Memberikan 1 *NPC* penjaga di dekat pintu menuju blok 2 menggunakan *mode event*, memberikan 3 *NPC* yang berkeliaran secara bebas.

2. Menjawab ada berapa total *NPC* di blok 1 dan 2 (Blok 2)
Nantinya setelah pemain melewati blok 1, pemain akan diberi pertanyaan oleh penjaga blok 2 berupa ada berapa total *NPC* yang ada di blok 1 dan blok 2



Gambar 2.12. *Level* Blok 2

Dengan memberikan 1 *NPC* penjaga di samping pintu menuju blok 3 menggunakan *mode event* dan memberikan 3 *monster* yang akan mengejar pemain jika pemain salah dalam menjawab kuis dengan

menggunakan *mode event* juga, kemudian memberikan 2 *NPC* bebas untuk menjadi bahan kuis nantinya.

3. Menjawab ada berapa hewan di blok 3 (Blok 3)

Jika sudah melewati blok 1 dan 2, di blok 3 akan ada banyak hewan berupa anjing, kucing, dan lain-lain. Nantinya pemain akan diberikan pertanyaan oleh penjaga berupa jumlah total hewan yang ada di blok 3 tersebut jika salah dalam menjawab pemain akan dikejar oleh sosok *monster* misterius tidak terlihat dan akan terlihat jika pemain menjawab pertanyaan dengan tidak benar.

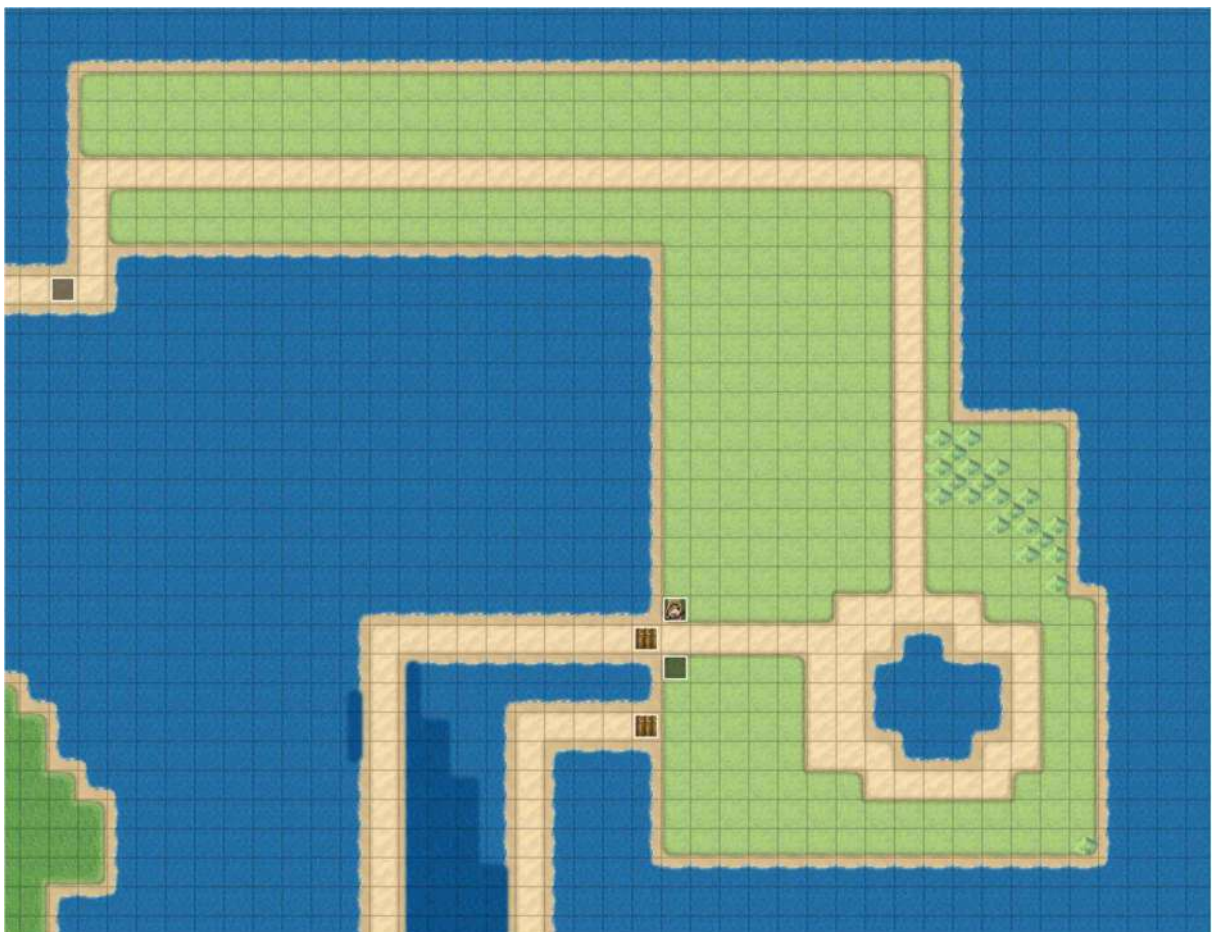


Gambar 2.13. *Level/ Blok 3*

Dengan memberikan 16 macam hewan ditempatkan secara acak menggunakan *mode event*.

4. Menjawab siapa nama adik Simo (Blok 4)

Di blok 4 ini pemain akan diberikan pertanyaan mengenai siapa nama adik dari pemain alias Simo, dimana nama adik pemain ini sudah dijelaskan di awal *game*.

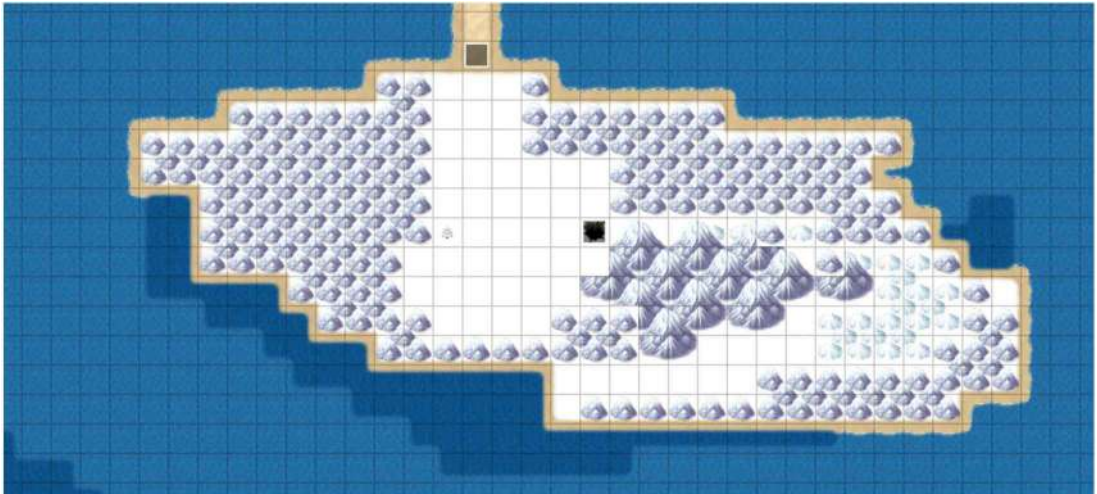


Gambar 2.14. *Level/ Blok 4*

Dengan memberikan *NPC* penjaga di depan pintu, dengan memberi gambar *actor* 3 untuk nanti memberikan kuis kepada pemain.

5. Bertemu dengan Somi di *bunker* (Blok 5)

Setelah pemain melalui 4 blok tersebut, akhirnya pemain tidak lagi melalui dengan menjawab kuis. Kini pemain hanya menemukan lubang dimana *bunker* tempat sembunyi Somi tersebut berada.



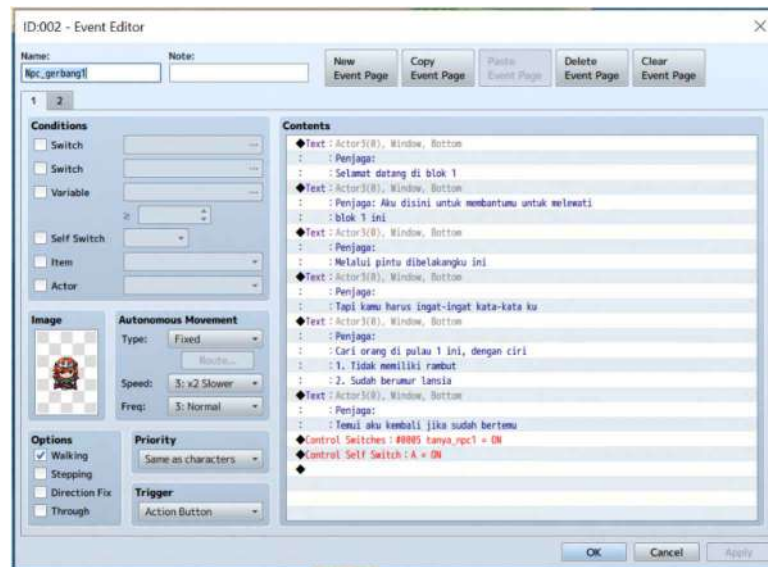
Gambar 2.15. Blok 5

Dengan memberikan lubang 1 menggunakan *mode event*, kemudian memberikannya gambar berupa lubang yang berada di *tileset c*.

Scripting

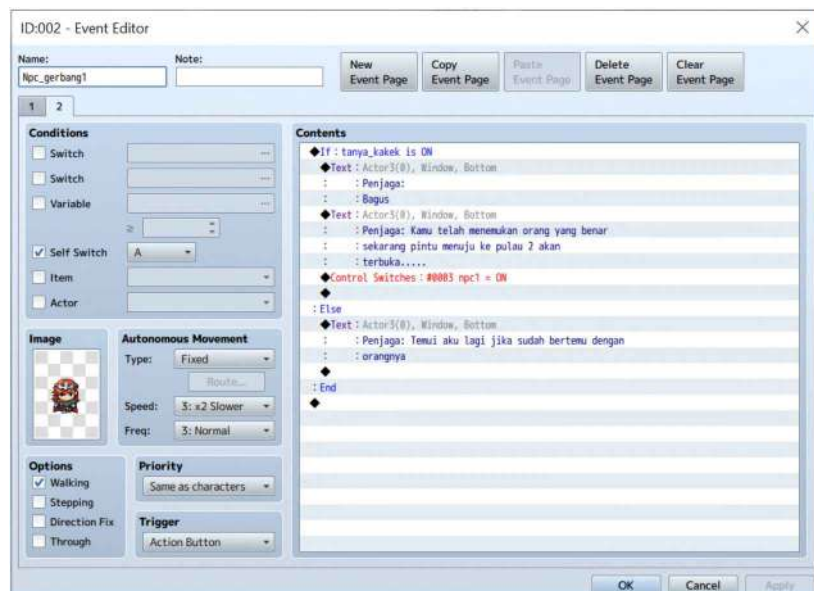
Scripting di RPG Maker MV tidak perlu menggunakan bahasa pemrograman. Berikut dibawah ini tahapan *scripting* di setiap *level* blok.

1. Blok 1



Gambar 2.16. *Scripting Penjaga Blok 1*

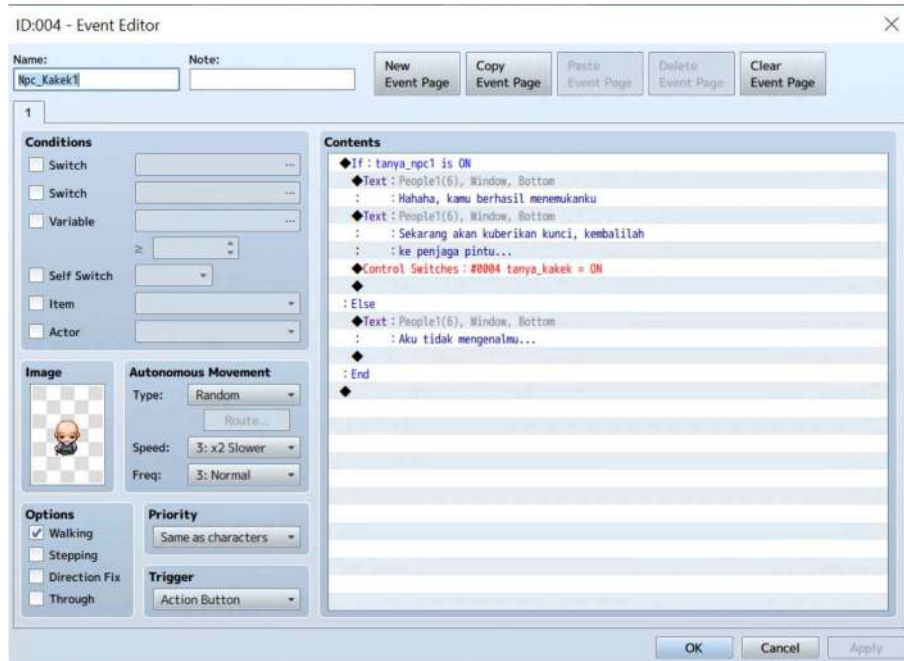
Dimulai dengan memberi kuis kepada pemain berupa mencari *NPC* yang diinginkan di blok 1 tersebut. Jika sudah memberi kuis otomatis penjaga langsung berganti ke *page 2* dibawah dengan menekan *new event page* diatas.



Gambar 2.17. Halaman 2 *scripting* penjaga

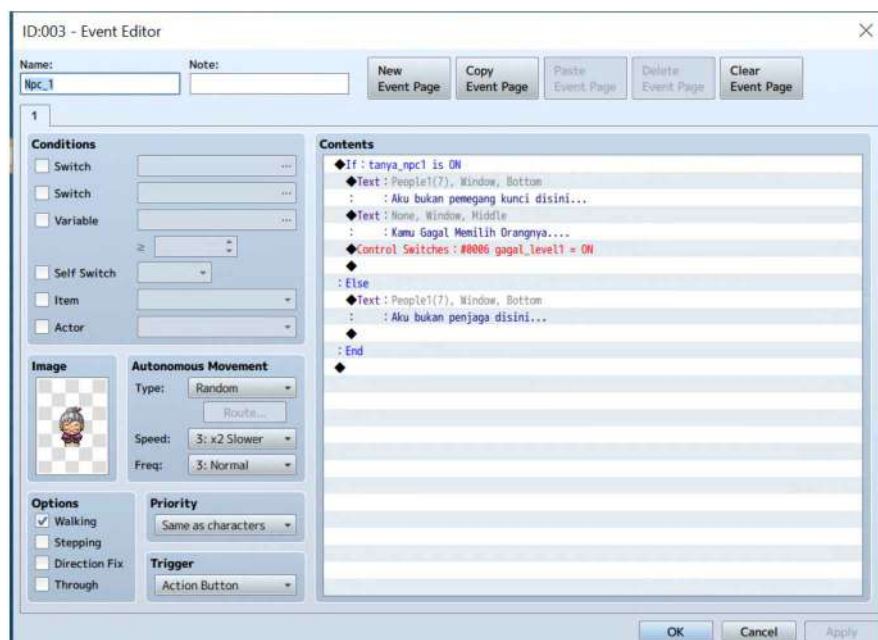
Ini adalah halaman kedua dari si penjaga, digunakan supaya tidak terjadi pengulangan kuis jika pemain berinteraksi dan akan mengaktifkan pintu terbuka jika

pemain sudah menemukan *NPC* yang dimaksud, dengan cara memakai *control switch*.



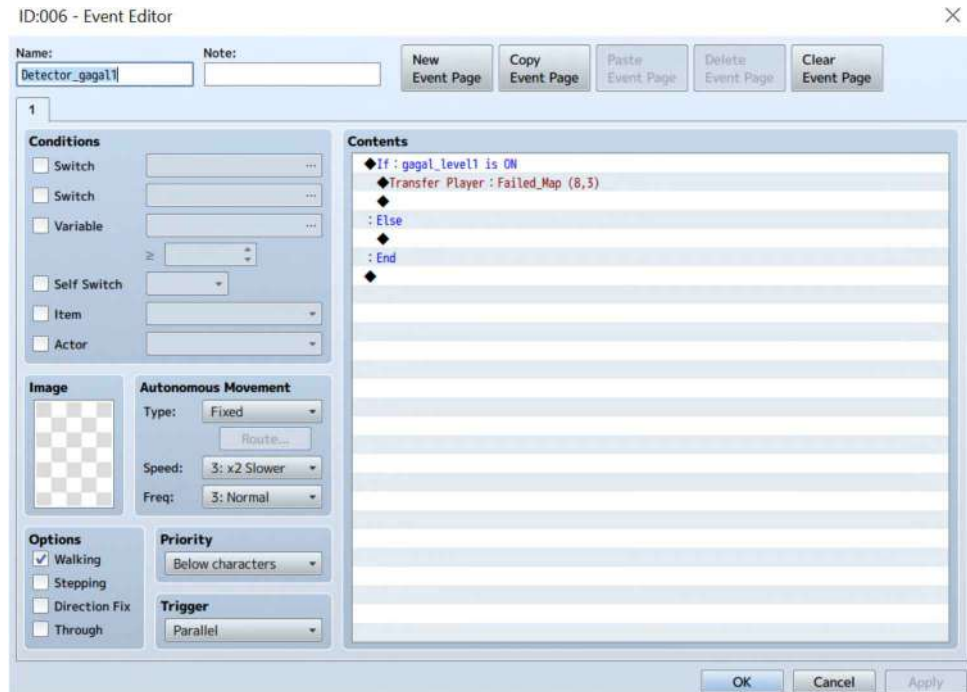
Gambar 2.18. Scripting NPC Objektif

Jika pemain berhasil menemukan *NPC* ini dan kembali ke penjaga maka otomatis penjaga akan mengetahuinya karena didalam *NPC* ini saya memberi *control switch* jika pemain menemukannya.



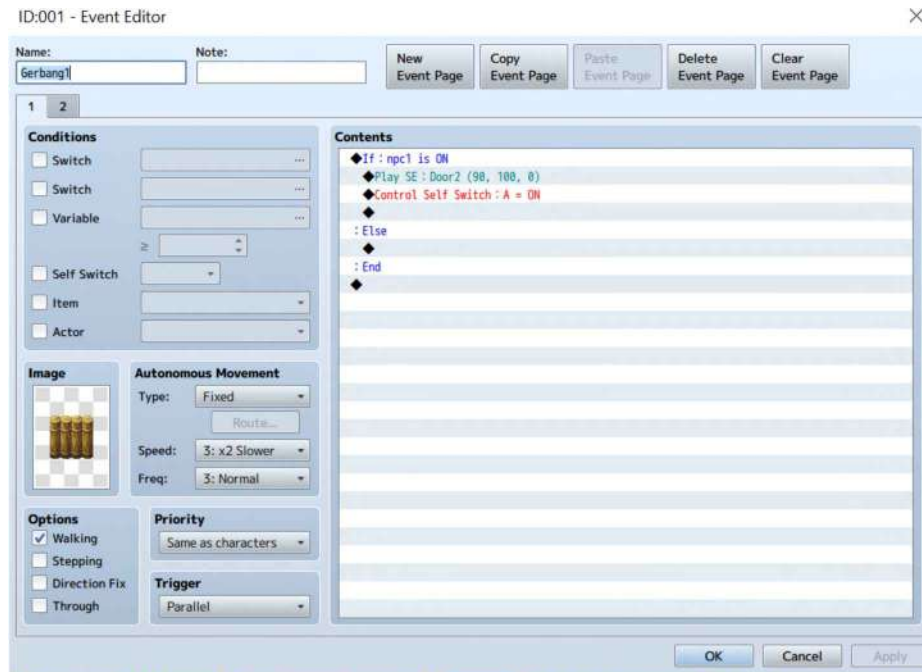
Gambar 2.19. Scripting NPC Bukan Objektif

Dengan memberinya *control switch* lagi jika pemain menemukan ini, untuk nanti terdeteksi otomatis pemain salah dalam mencari *NPC*.



Gambar 2.20. *Detector* Gagal Blok 1

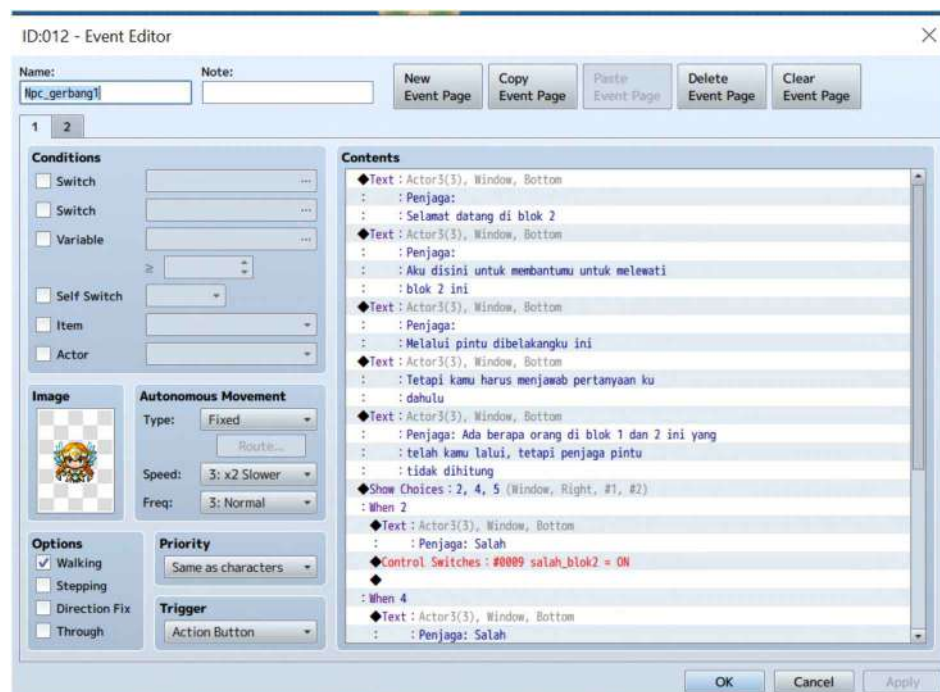
Dibuat dengan *trigger parallel* karena bersifat mendeteksi terus menerus. Jika *trigger* nyala maka *detector* ini akan menjalankan *script* di atas yaitu membawa pemain ke *screen game over*.



Gambar 2.21. Scripting Pintu Blok 1

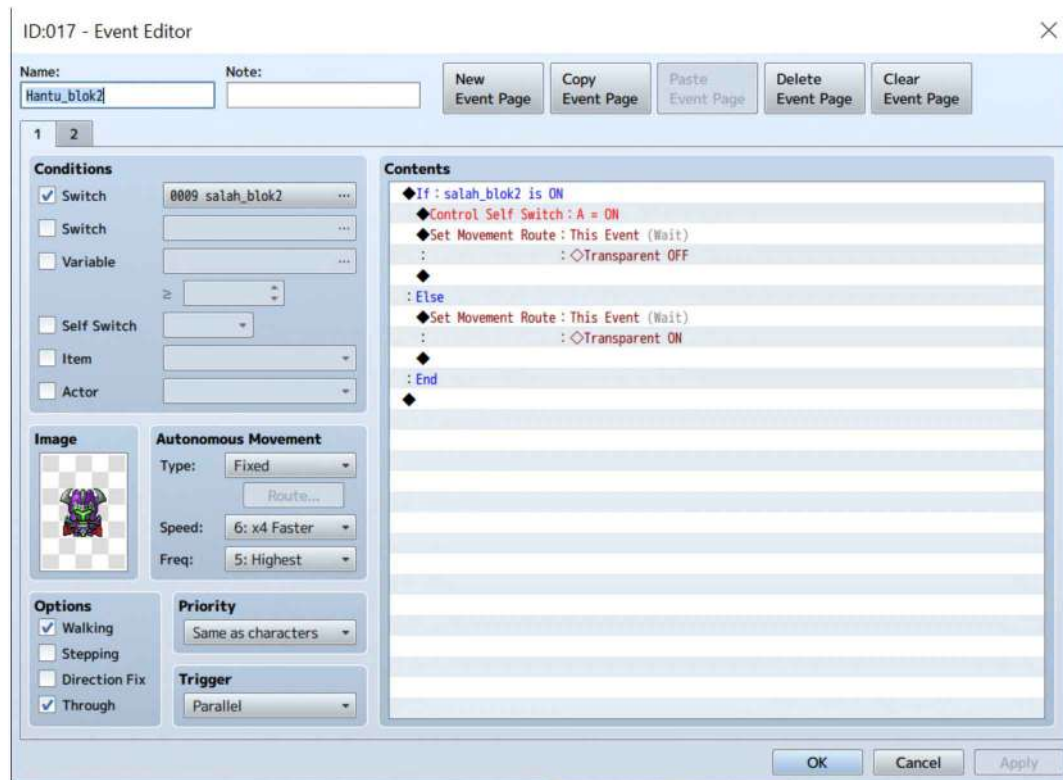
Jika pemain sudah menemukan *NPC* yang benar maka pintu ini akan otomatis terbuka dan pemain bisa pergi ke blok 2 karena *trigger*-nya bersifat *parallel*.

2. Blok 2



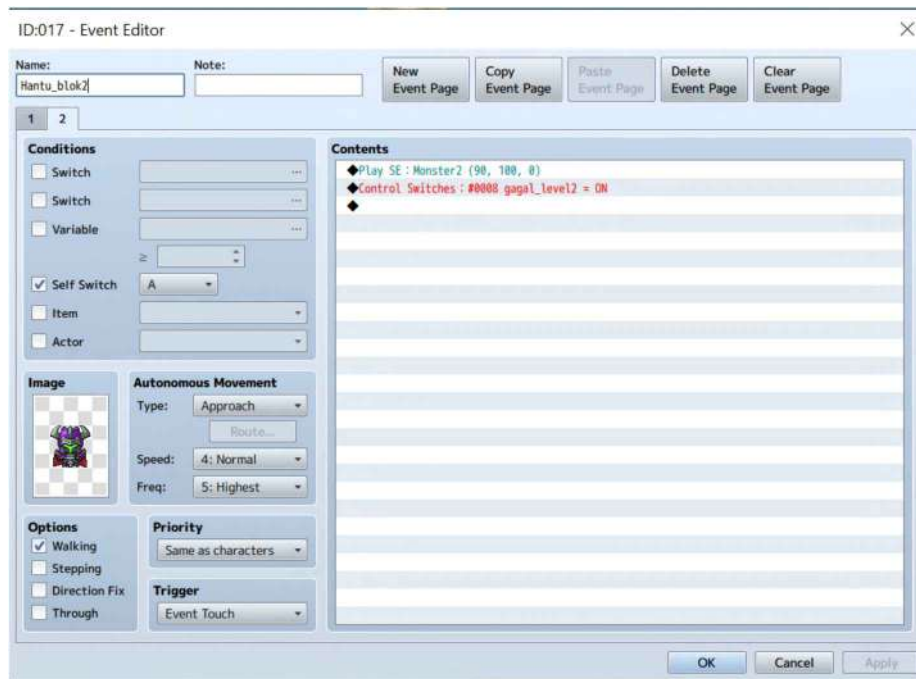
Gambar 2.22. Scripting Penjaga Blok 2

Disini hanya memberi pertanyaan menggunakan opsi *show choice* jika pemain salah dalam menjawab maka akan mengaktifkan *trigger monster* di bawah.



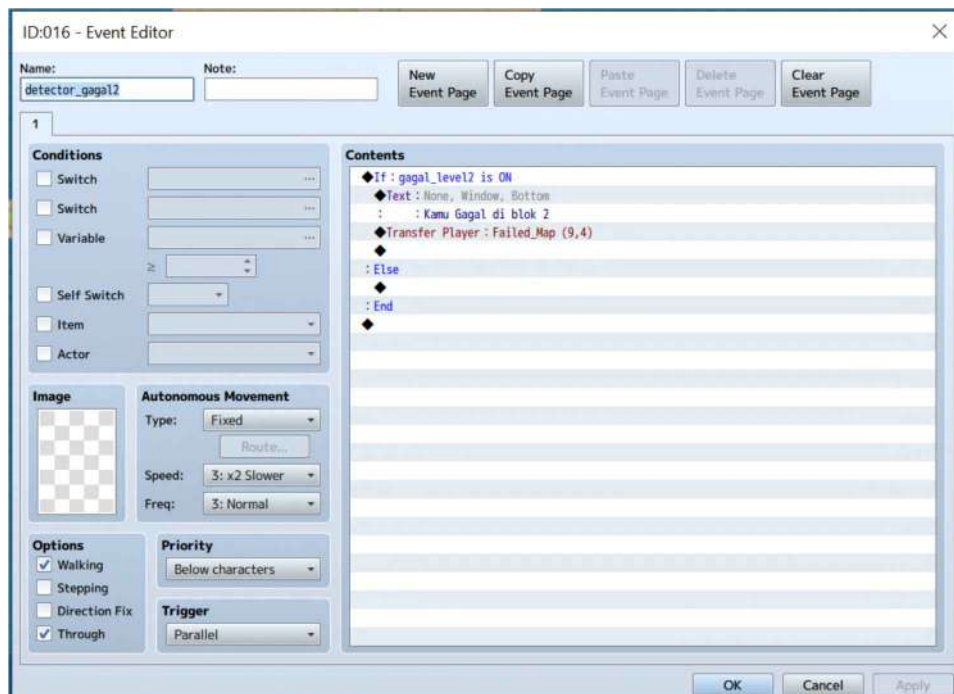
Gambar 2.23. *Script Monster*

Monster ini akan aktif dan terlihat jika *trigger* “salah” menjadi aktif atau pemain menjawab pertanyaan salah dimana ini berada di halaman *scripting* penjaga diatas dan *monster* ini akan berubah mengejar pemain.



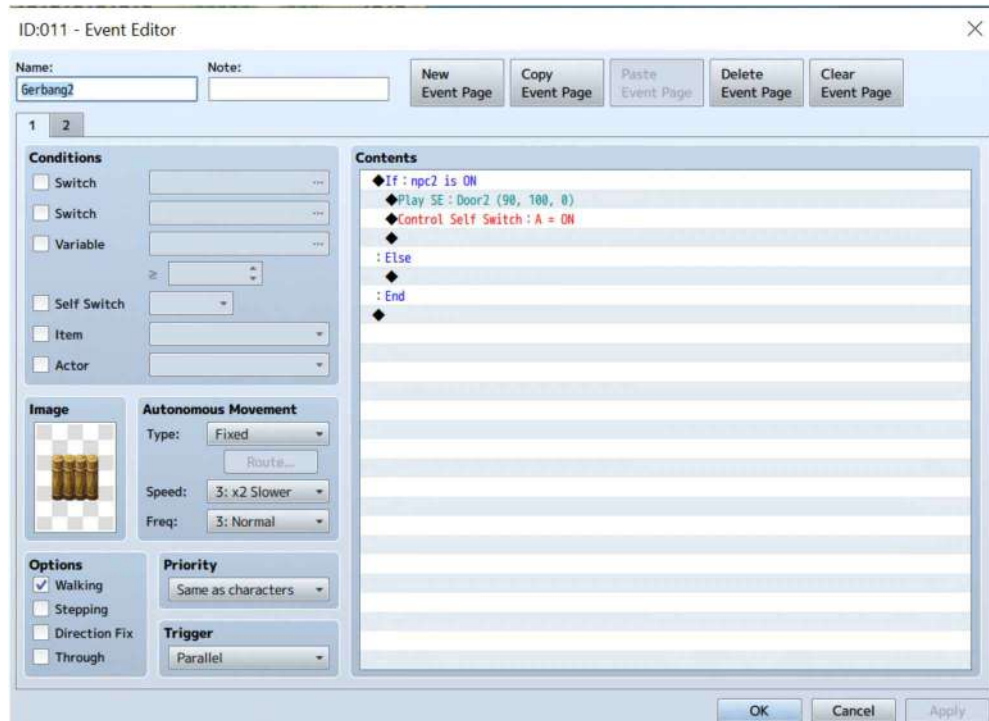
Gambar 2.24. Halaman 2 *Script Monster*

Disini *trigger*-nya *event touch* yang artinya jika tersentuh pemain maka akan mengaktifkan *trigger* untuk *game over*.



Gambar 2.25. *Detector Gagal Blok 2*

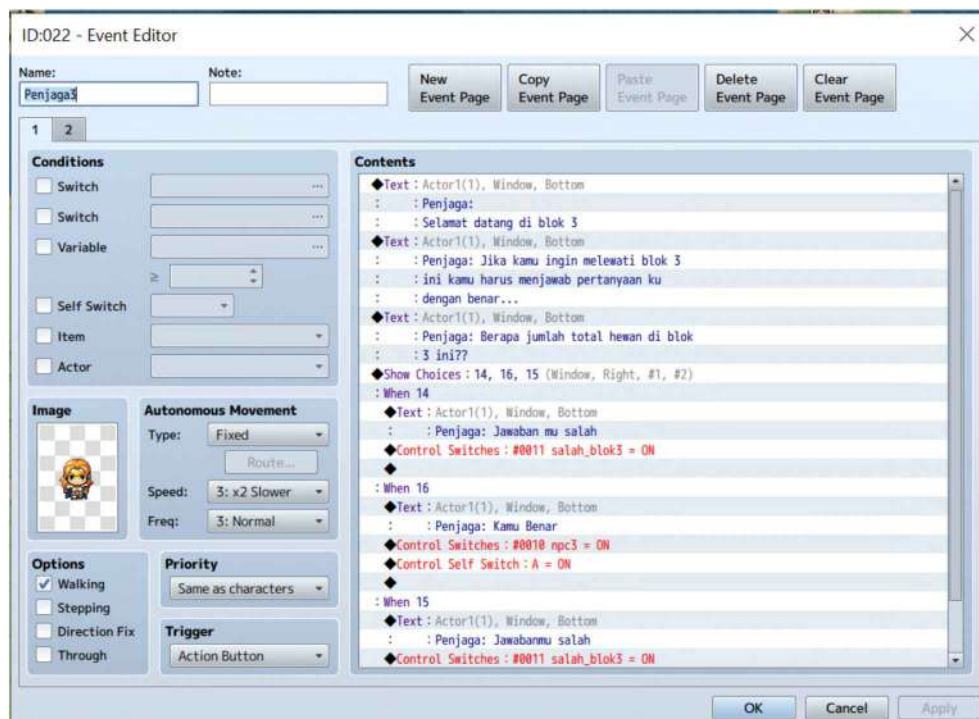
Ini akan aktif otomatis karena bersifat *parallel*, nantinya jika aktif akan membawa pemain ke *game over screen*.



Gambar 2.26. Script Pintu Blok 2

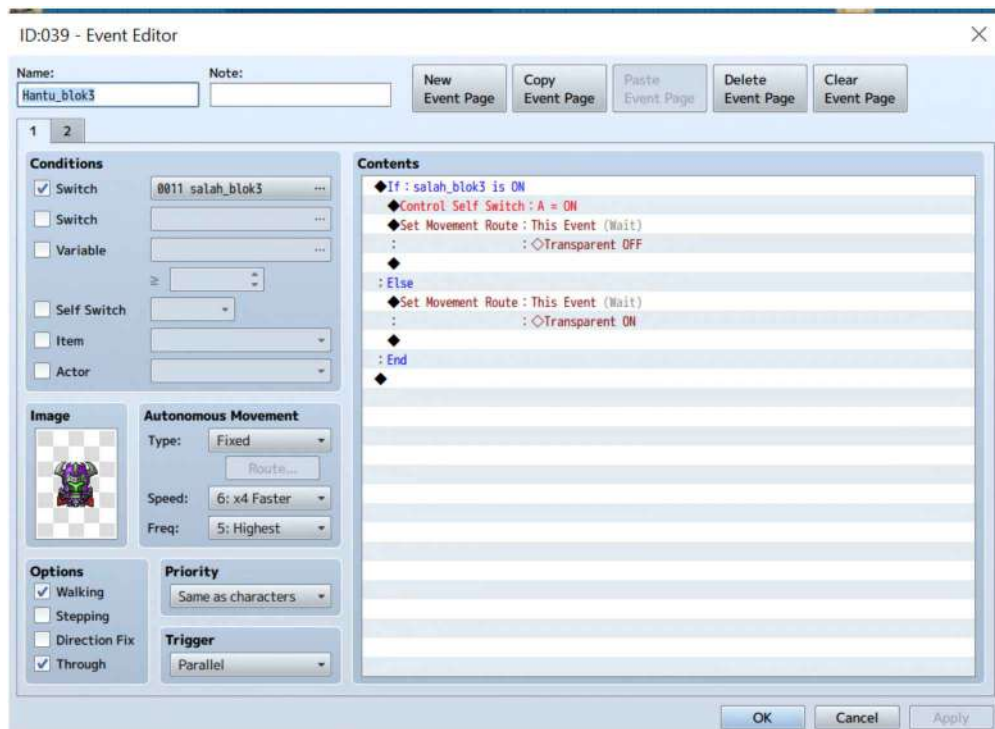
Pintu akan otomatis terbuka karena bersifat *parallel*, jika pemain menjawab dengan benar pertanyaan penjaganya.

3. Blok 3



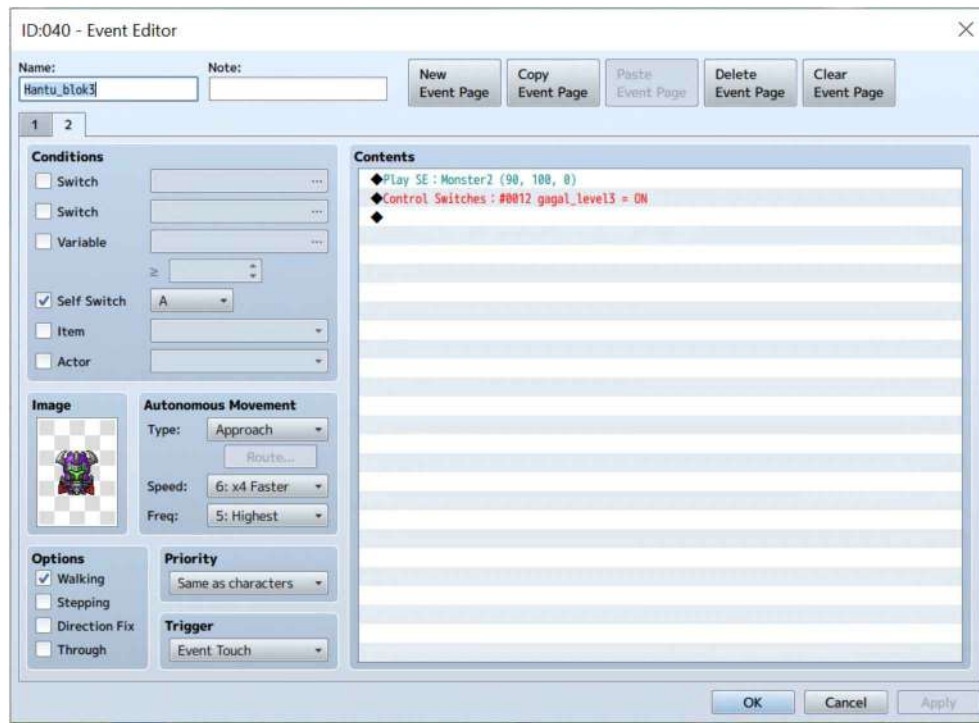
Gambar 2.27. Script Penjaga Blok 3

Disini menggunakan *show choice* lagi yaitu menebak jumlah hewan yang berada dalam blok 3. Jika salah maka akan mengaktifkan *switch* salah, tetapi jika benar maka akan mengaktifkan *switch* benar.



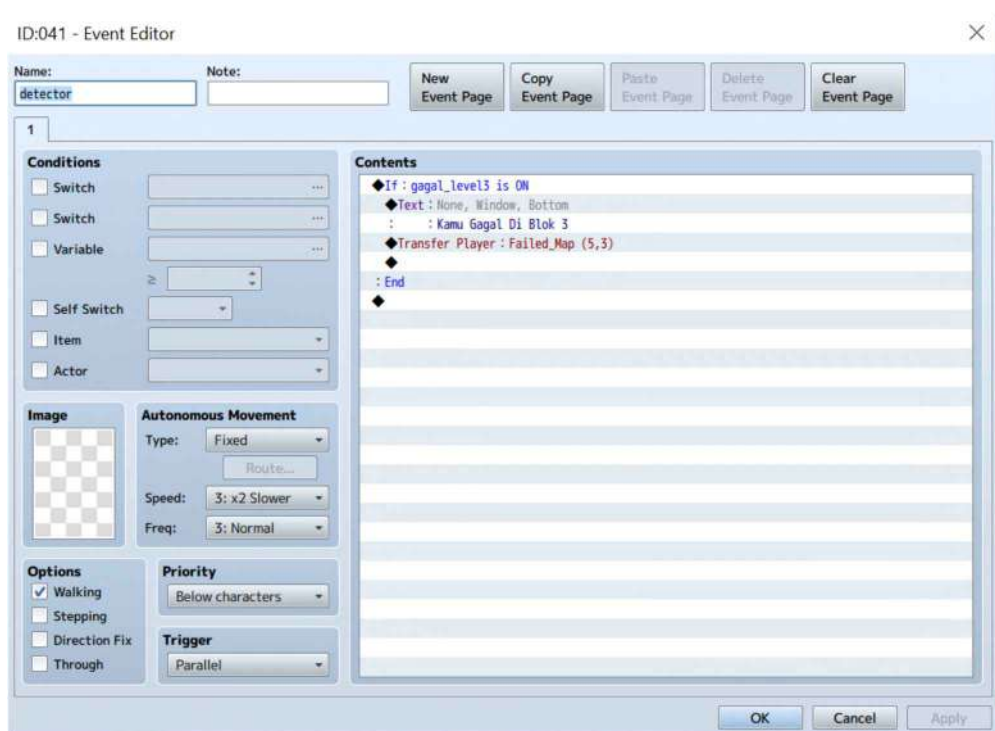
Gambar 2.28. *Script Monster*

Ini akan aktif dan terlihat jika *switch* salah menyala setelah itu akan mengejar pemain.



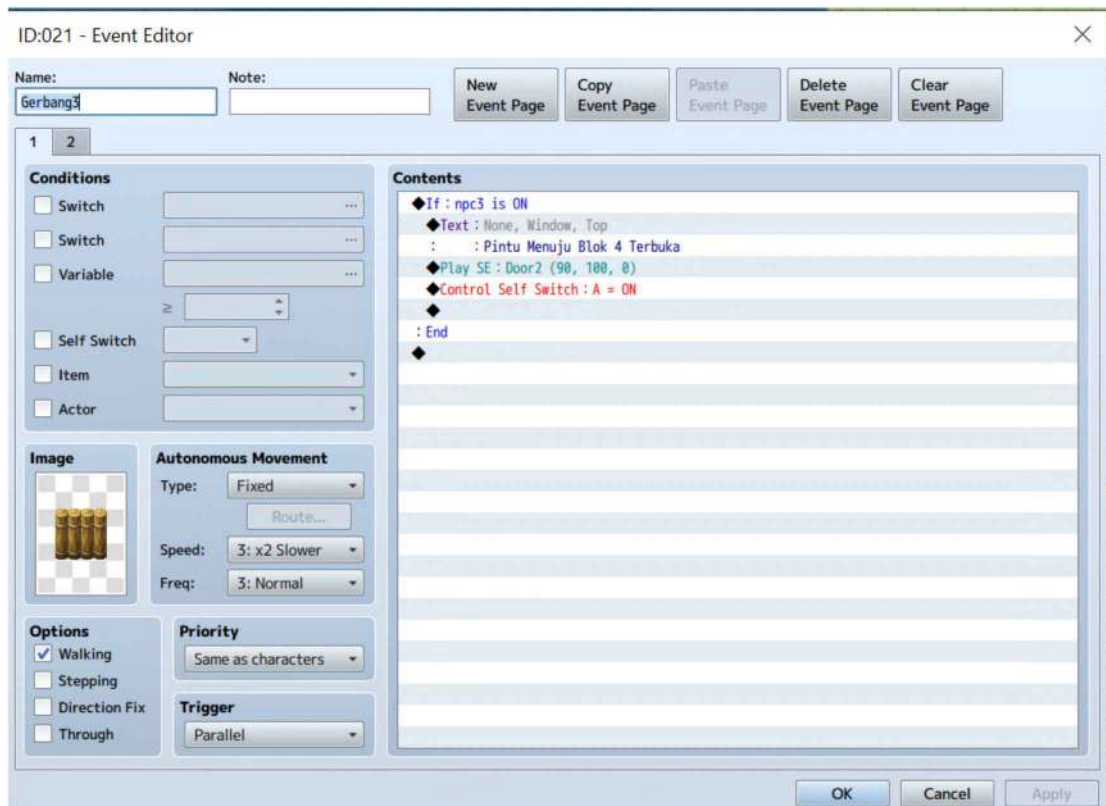
Gambar 2.29. Halaman 2 *Script Monster*

Disini *monster* akan mengejar pemain dan jika mengenai pemain maka akan mengaktifkan *switch* gagal.



Gambar 2.30. *Detector Gagal Blok 3*

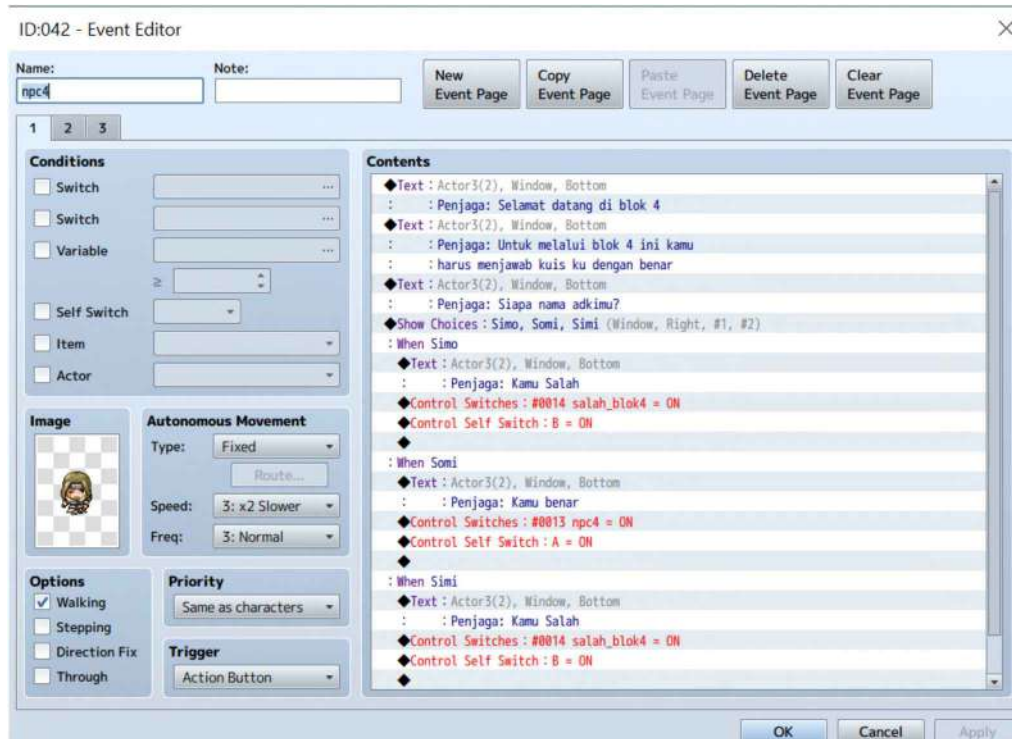
Ini akan otomatis aktif jika *switch* gagal nyala, yaitu akan membawa pemain ke *screen game over*.



Gambar 2.31. Script Pintu Blok 3

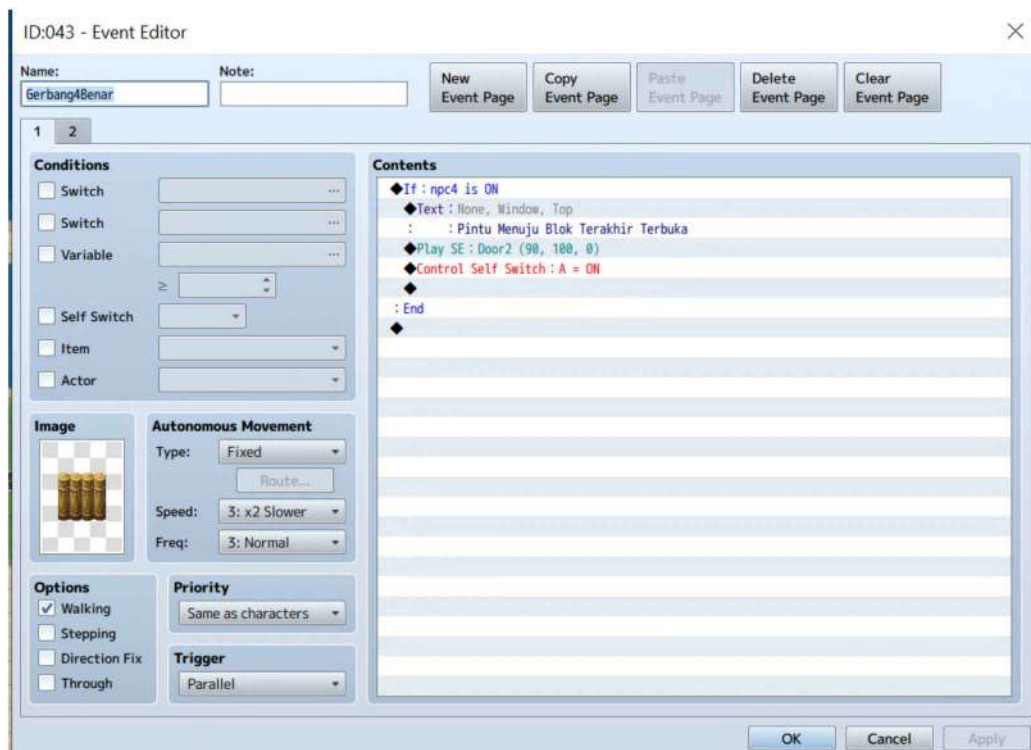
Ini akan aktif otomatis juga karena menggunakan *trigger parallel*, akan langsung terbuka jika pemain menjawab dengan benar berapa jumlah hewannya. Ini menggunakan *conditional branch* untuk *if else*-nya.

4. Blok 4



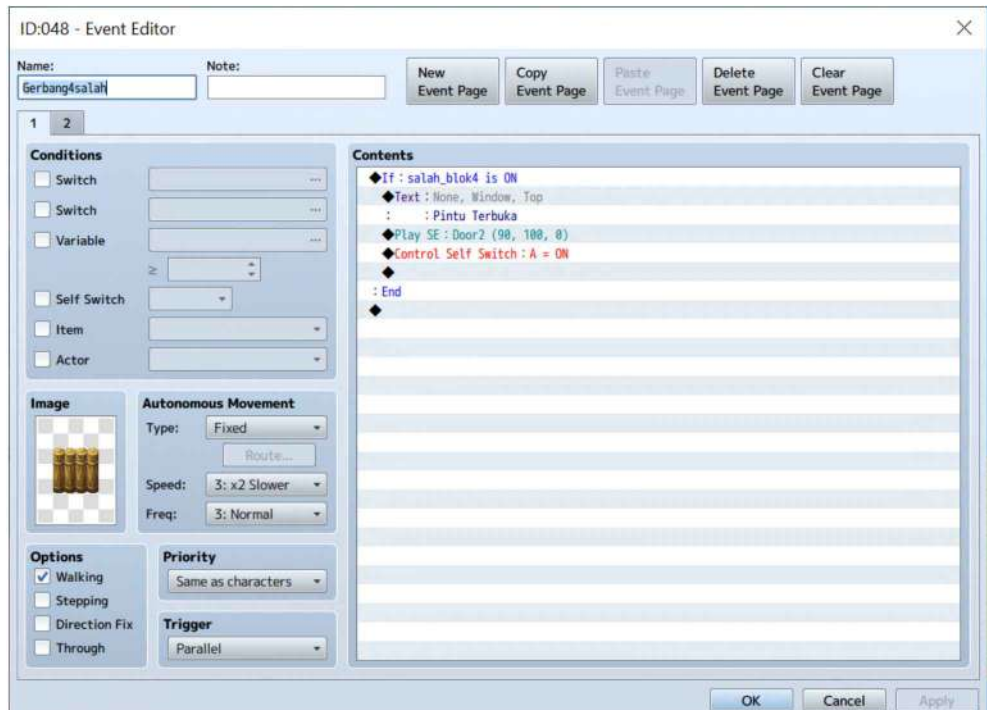
Gambar 2.32. *Script Penjaga Blok 4*

Disini akan memberi pertanyaan kepada pemain menggunakan *show choice* juga jika benar maka akan mengaktifkan *switch* "benar" dan sebaliknya.

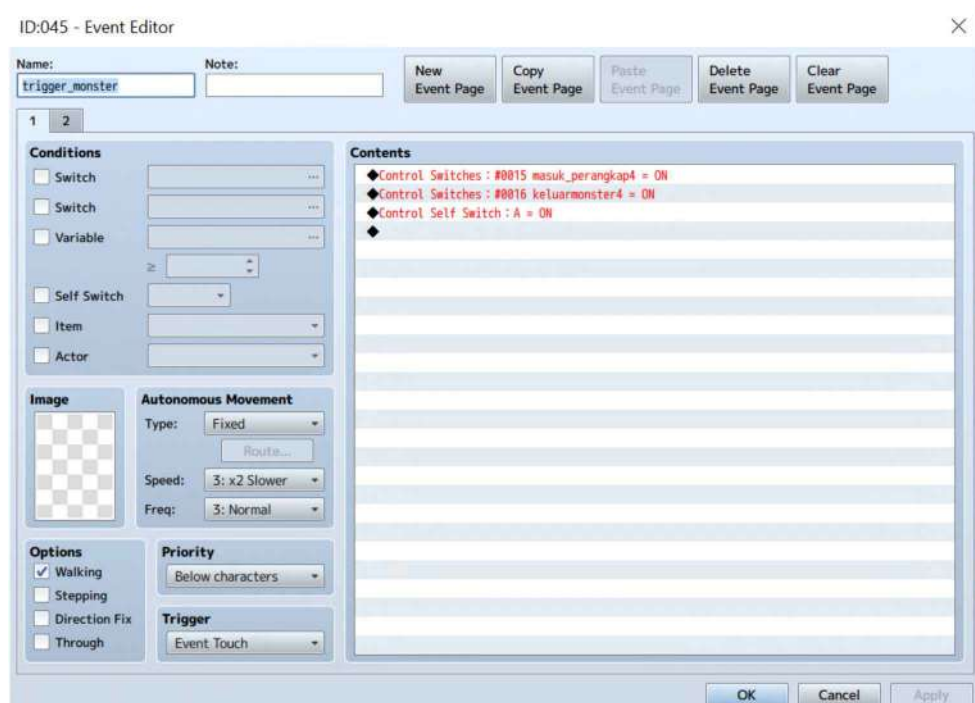


Gambar 2.33. *Script Pintu Yang Benar*

Menggunakan *trigger parallel*, ini akan terbuka otomatis jika pemain menjawab dengan benar.

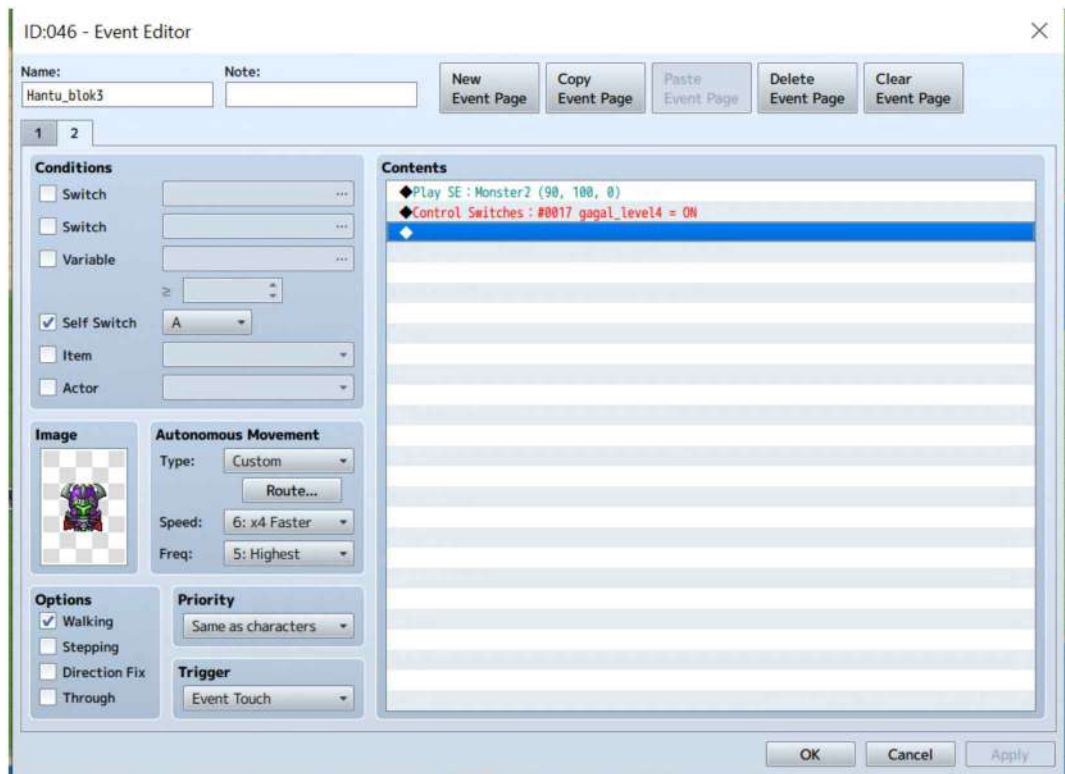


Gambar 2.34. *Script Pintu Salah*
Pintu salah ini akan terbuka dan pintu yang benar tetap tertutup karena menggunakan *conditional branch*.



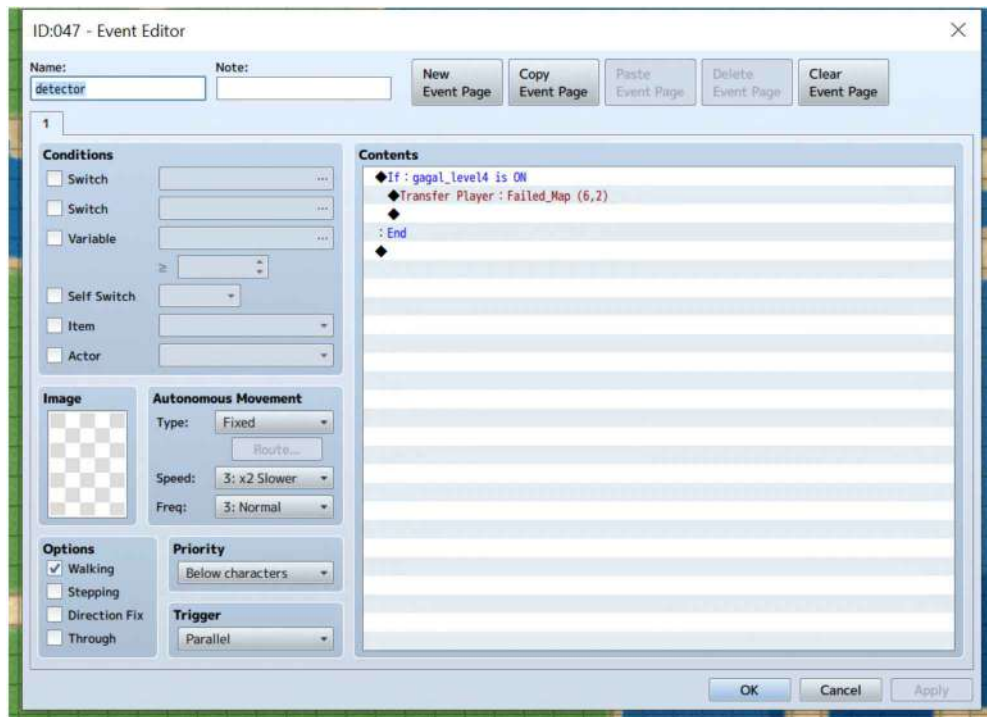
Gambar 2.35. *Script Jebakan*

Script ini menggunakan *trigger event touch*, jika dilalui pemain maka akan mengaktifkan *monster-monster* dan menutup jalan yang di belakang supaya pemain tidak bisa kembali.



Gambar 2.36. *Script Monster*

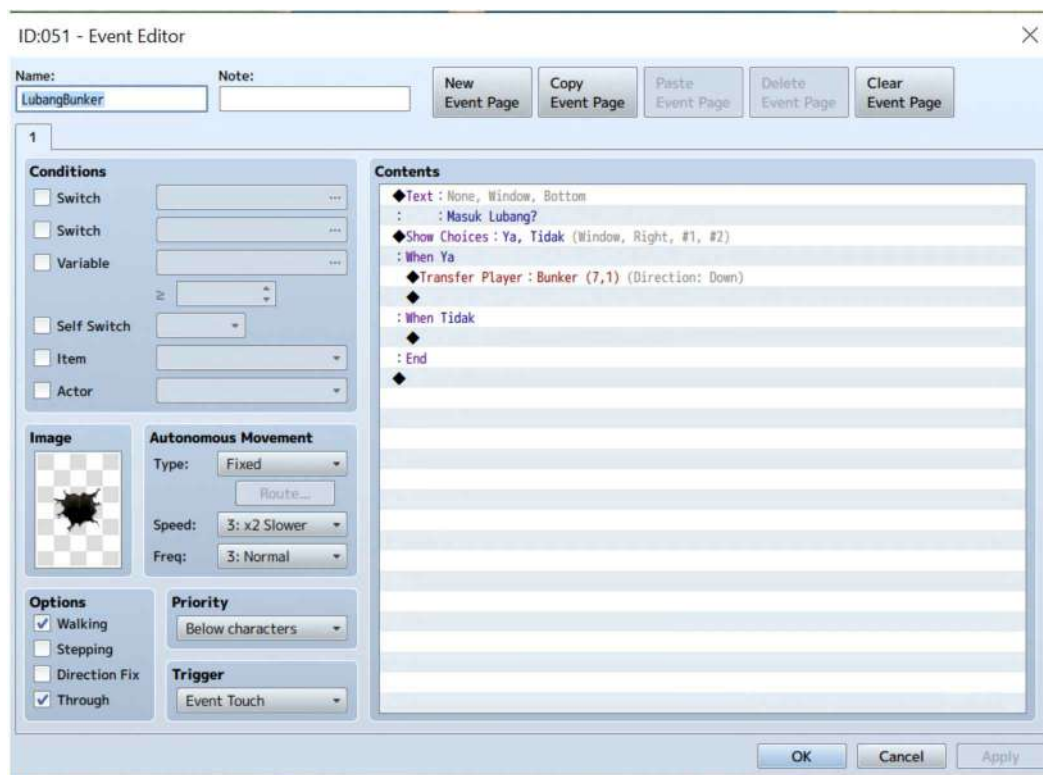
Menggunakan *trigger event touch* jika mengenai pemain akan mengaktifkan *switch gagal*.



Gambar 2.37. Script Detector Gagal Blok 4

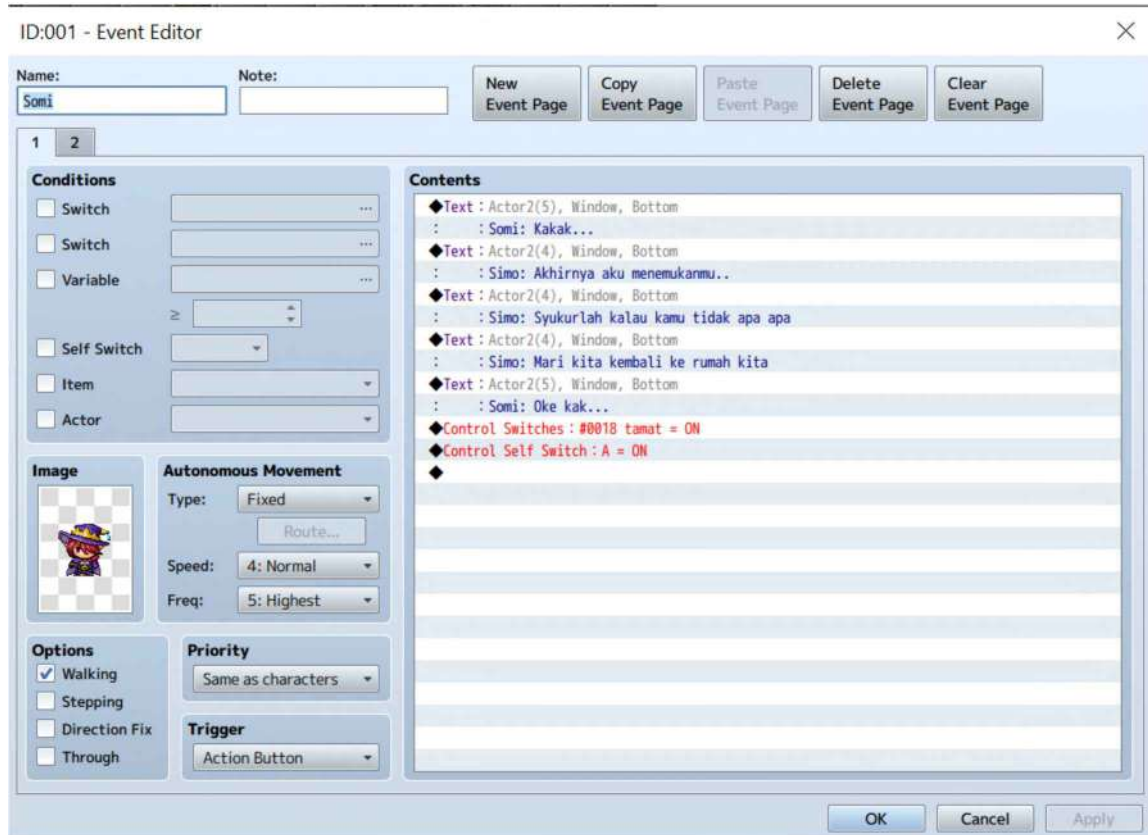
Ini menggunakan *trigger parallel* dan akan membawa pemain ke *screen game over* jika *switch* gagal menyala.

5. Blok 5



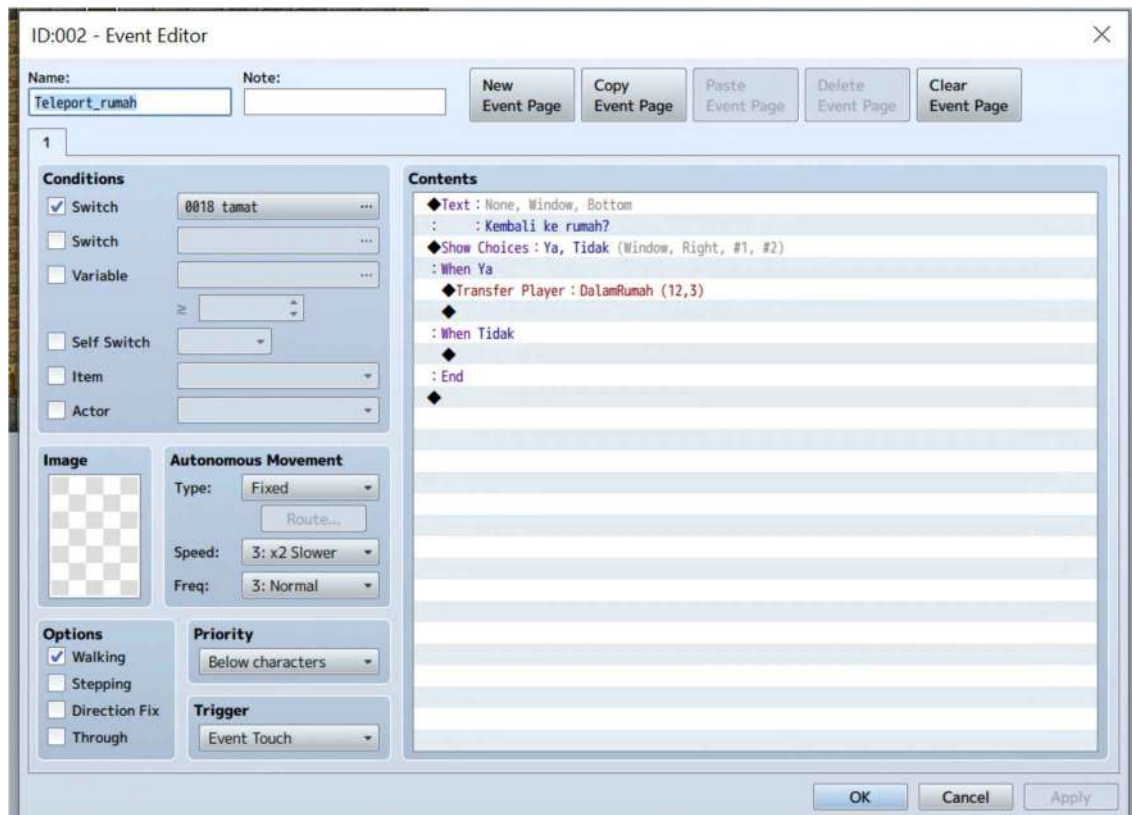
Gambar 2.38. Script Lubang

Menggunakan *trigger event touch* karena jika dilalui pemain akan memberi pilihan untuk masuk lubang atau tidak dengan menggunakan *show choice* dan *transfer player*.



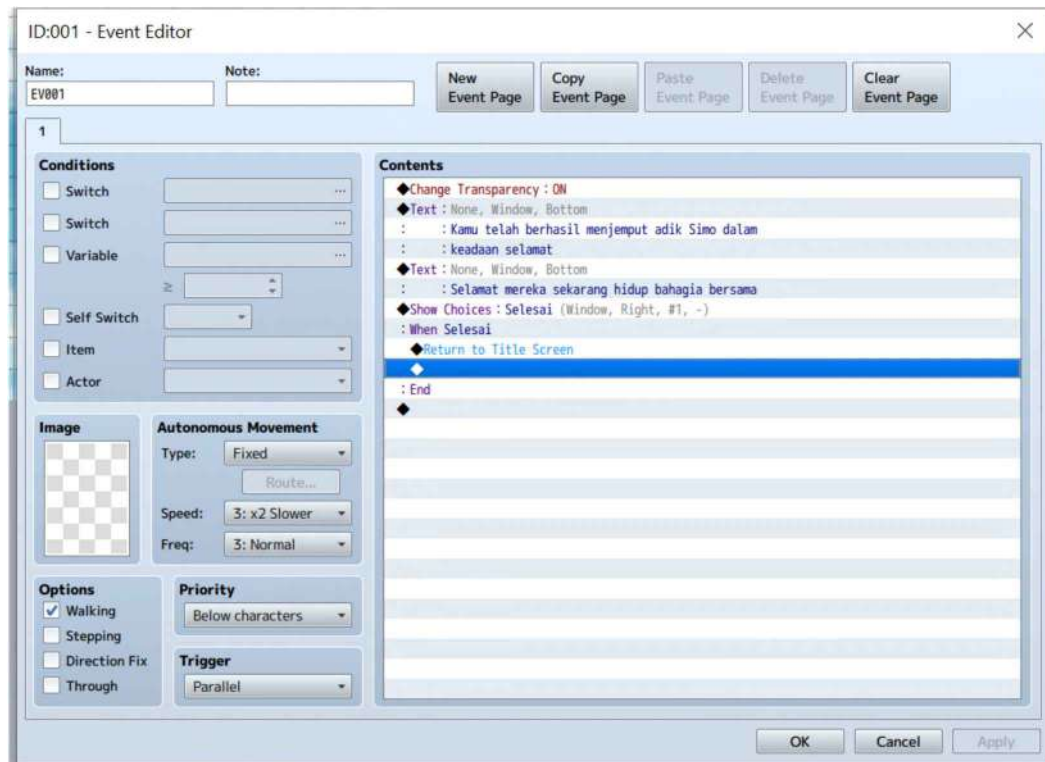
Gambar 2.39. Script Somi

Disini menggunakan *show text* untuk mengobrol setelah itu akan mengaktifkan *switch* “tamat”.



Gambar 2.40. Script Pintu Tamat

Ini akan muncul jika *switch* “tamat” menyala. Ini menggunakan *trigger event touch* dan *show choice* jika pilih “ya” maka akan membawa pemain ke *screen* “tamat”.



Gambar 2.41. Script Tamat

Menggunakan *trigger parallel*, *show text*, *show choice*, *return to title screen*. Setelah beberapa percakapan maka akan kembali ke *main menu game*.

Ekspor Game

Ekspor *game* RPG Maker MV digunakan untuk menjadikan proyek *game* yang telah dibuat menjadi *game* yang benar-benar jadi dan bisa dimainkan. Berikut dibawah ini langkah-langkah ekspor *game* RPG Maker MV.

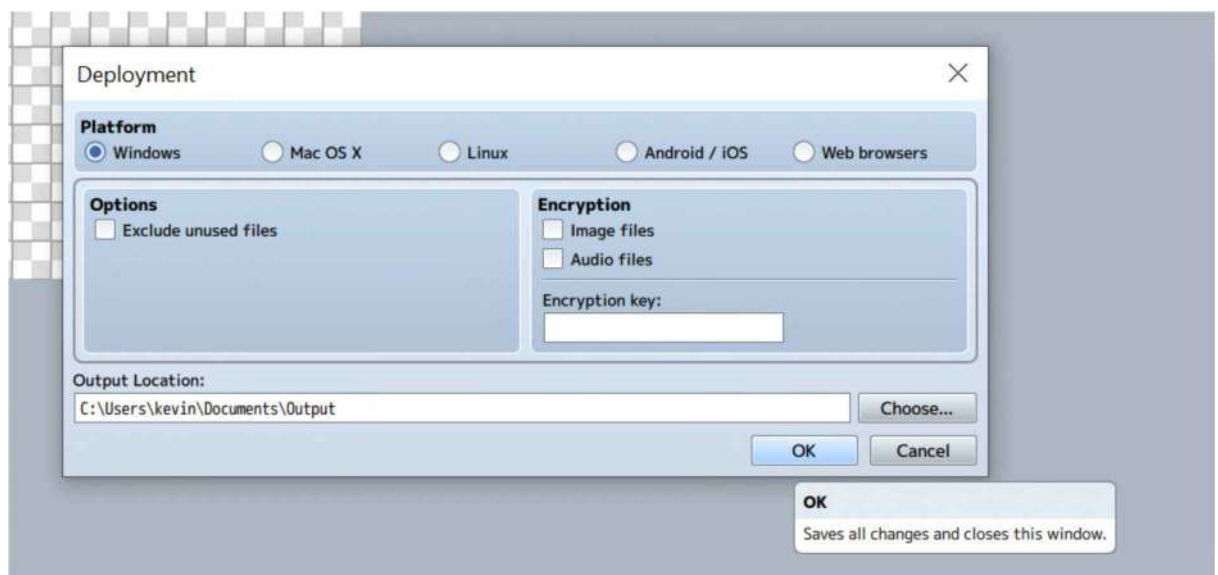
1. Pilih *Menu File*.



Gambar 2.42. *File Menu*

Menu *file* terdapat di pojok kiri atas, kemudian pilih *deployment*.

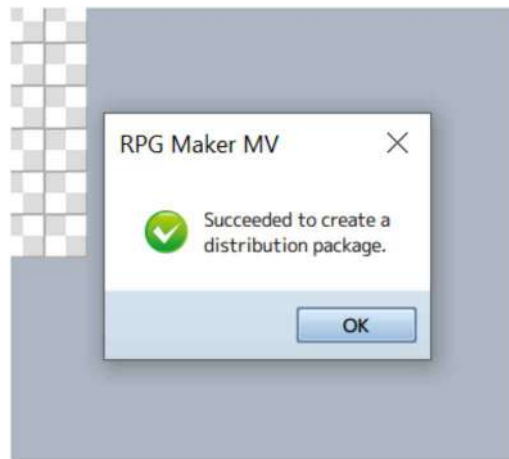
2. Pilih *platform* Windows, kemudian klik *ok*.



Gambar 2.43. *Deployment*

Ada beberapa pilihan *platform*, tetapi saya menggunakan yang Windows saja. Kemudian pilih *output location* yang diinginkan, saya sarankan dari *default*-nya saja di *document*.

3. Jika proses ekspor sudah selesai maka akan keluar seperti yang dibawah ini.



Gambar 2.44. Sukses Ekspor

Begitulah caranya mengekspor proyek menuju *game* RPG Maker MV yang siap dimainkan.

Link Video Gameplay Dan Link Download Game

Bagi yang ingin menonton video *gameplay* dari *game* ini bisa kunjungi *link* berikut: <https://youtu.be/MhW3UTZ9qGU> (Link Video, n.d.)

Berikut *link* yang bisa saya *share*, bagi yang ingin mendownload *game*-nya:

- *Link* 1 (<https://tinyurl.com/pjyb1>)
- *Link* 2
(<https://drive.google.com/file/d/14JqUQfpYlwbpUsPbMviKaanj24KMk1g1/view?usp=sharing>)

Kota Yang Terbangkalai

Adinata Ekwan Januar Pratama

Terdapat tiga karakter yang bernama Kenta, Izzi, dan Dicky yang tidak sengaja terdampar di kota kecil. mereka kemudian melakukan percakapan dengan penduduk kota itu dan tidak disangka bahwa mereka adalah orang yang dimaksud dalam ramalan Dukun di kota tersebut nantinya akan mengalahkan *monster* itu dan menghidupkan kembali kota tersebut.

Tampilan Keseluruhan Map



Gambar 3.1. *Map Snow Town* sebagai utama dalam *game* ini



Gambar 3.2. Tampilan *Map Bar*



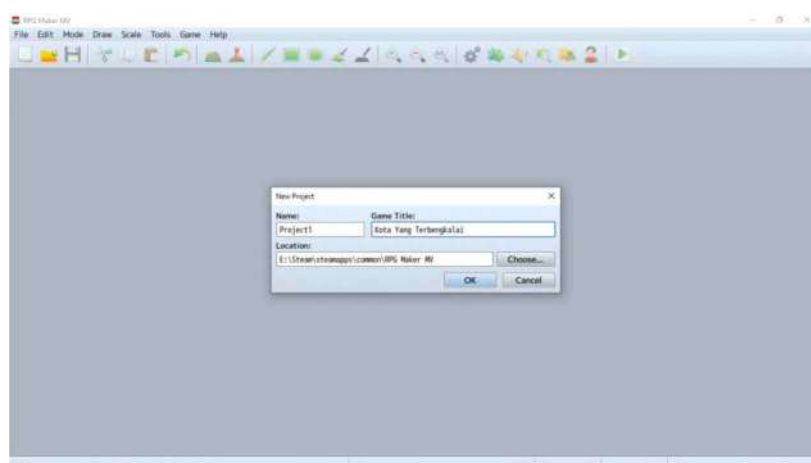
Gambar 3.3. Tampilan *Map Armor Shop*



Gambar 3.4. Tampilan *Map* Rumah *Monster*

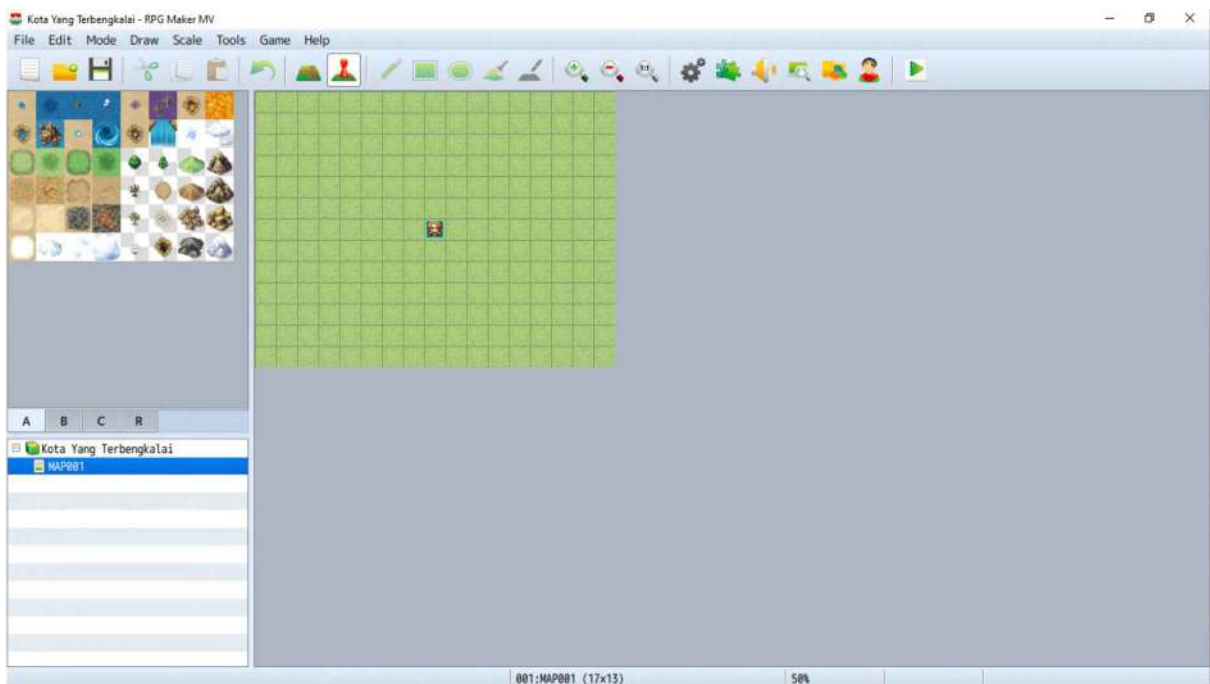
Memulai RPG Maker MV

Setelah membuka *software* RPG Maker MV, kemudian klik *File* -> *New Project* atau Ctrl+N untuk membuat *project*. Kemudian tulis nama, judul *game*, dan pilih lokasi penyimpanan lalu klik OK.



Gambar 3.5 Membuat *file project* baru

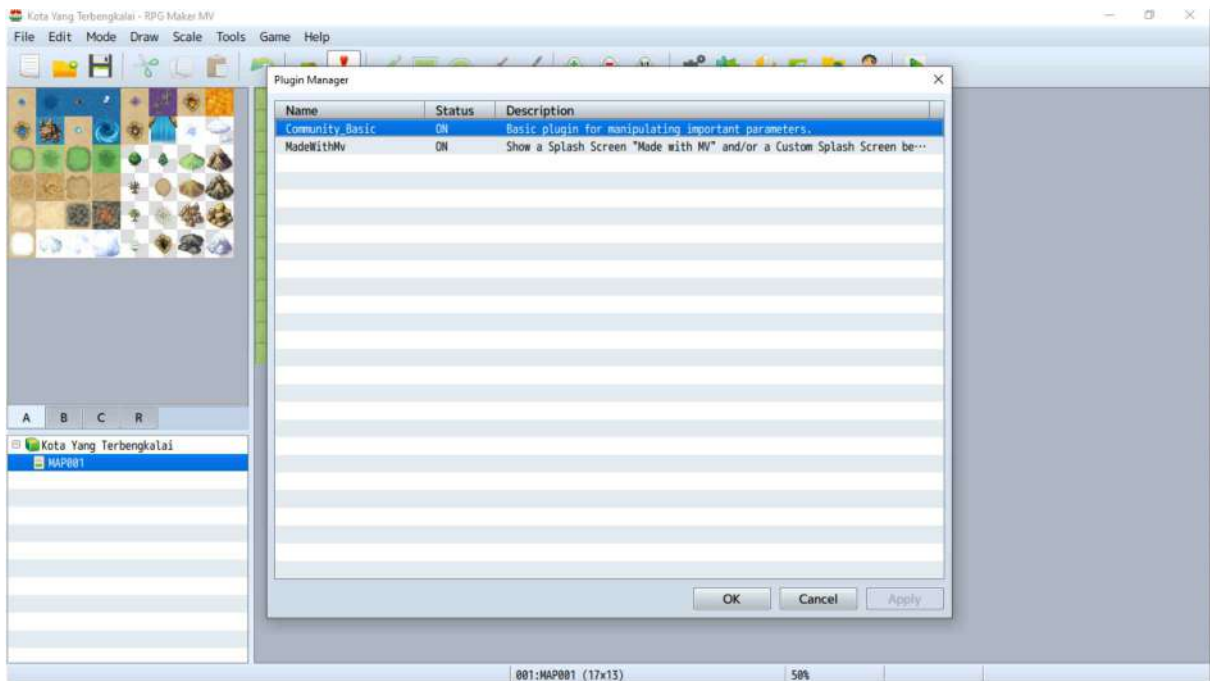
Tunggu hingga *loading* selesai. Setelah *loading* selesai maka akan muncul tampilan seperti di bawah ini.



Gambar 3.6. Tampilan setelah *loading file* baru

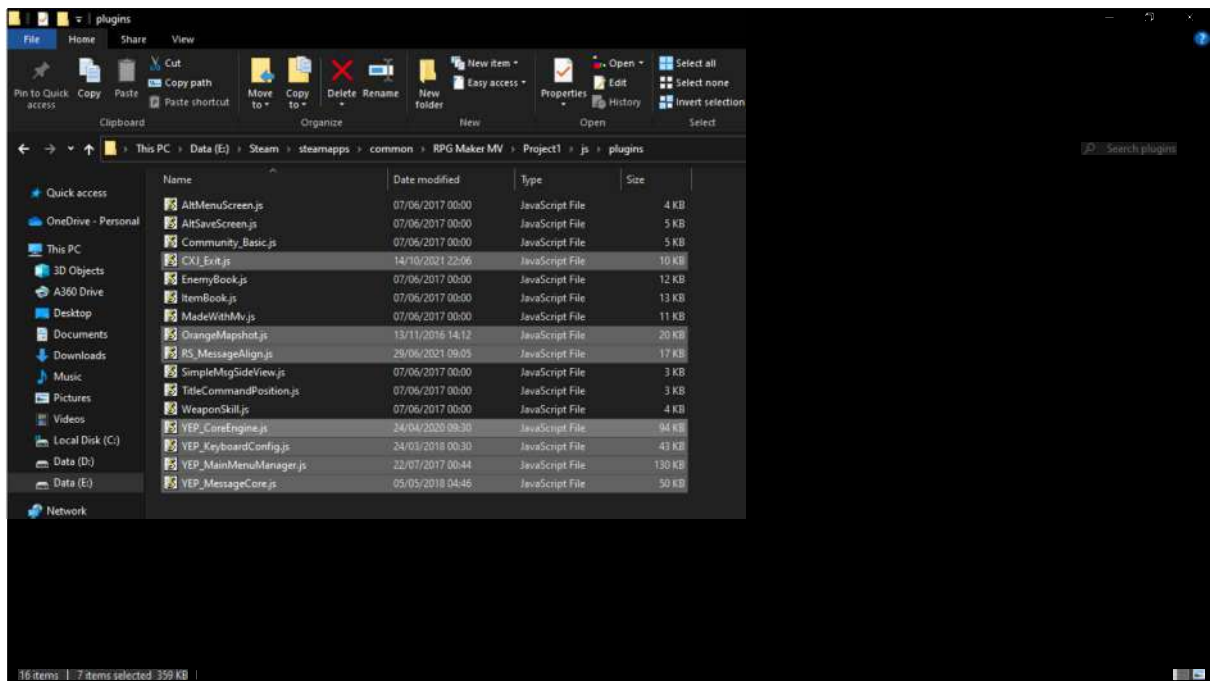
Meng-install Plugin

Sebelum memulai membuat *game*, *install plugin* terlebih dulu. *Plugin* adalah sebuah program berbasis *Javascript* untuk menambah fitur tertentu. Cara membuka menu *plugin* adalah dengan menuju menu *tools* → *Plugin Manager* atau menekan tombol F10.



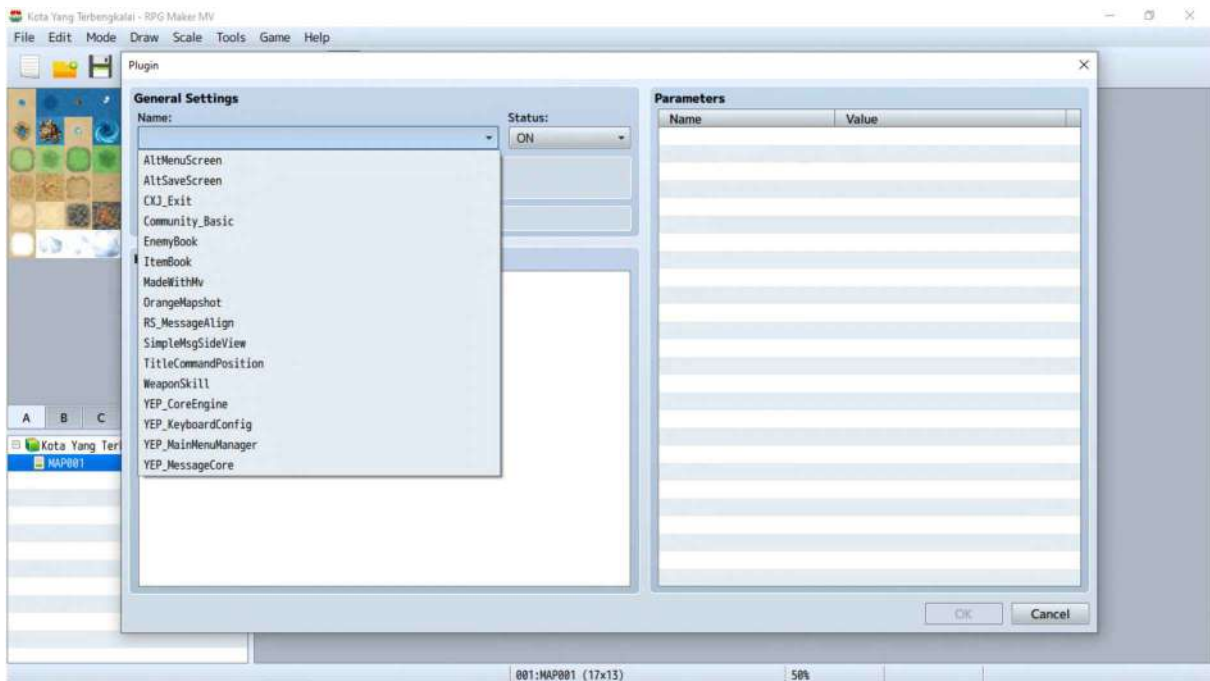
Gambar 3.7. Tampilan menu *Plugin*

File plugin dapat dilihat di *file* dokumen *game* → *js* → *plugin*. Anda bisa *download plugin* yang telah saya sediakan di <https://s.id/KotaYangTerbengkalai>. Setelah itu *copy* dan *paste* di *folder plugins* tersebut. Hasilnya akan seperti di bawah ini.



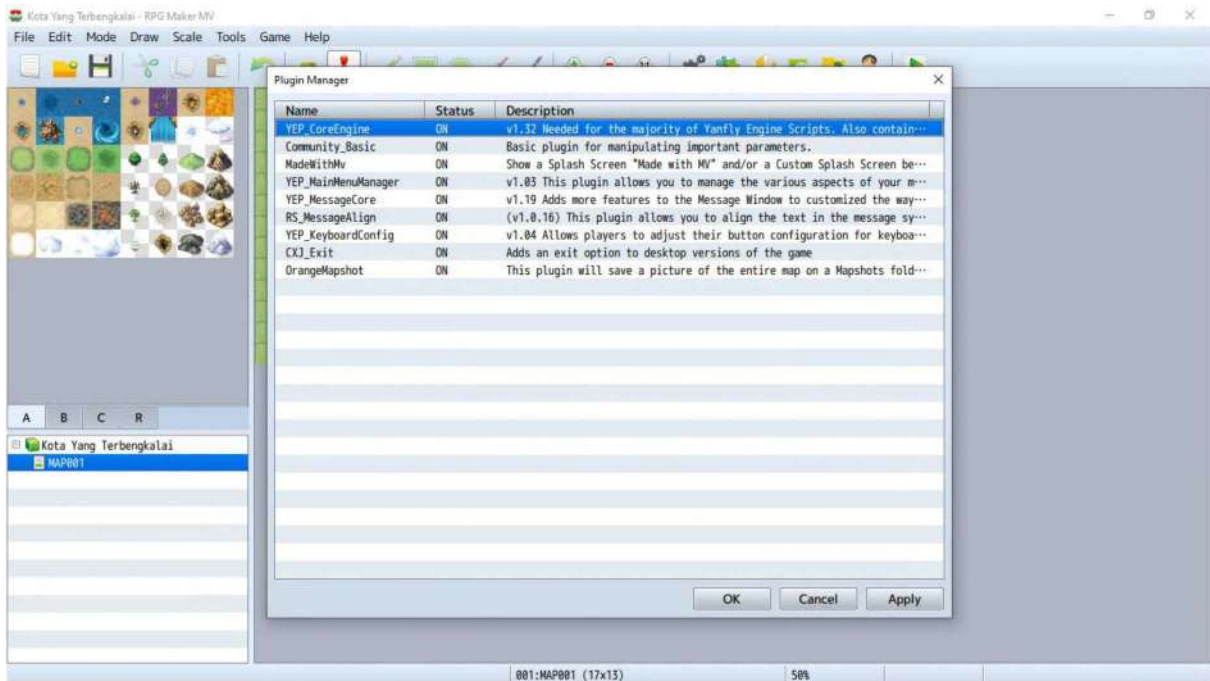
Gambar 3.8. Folder *Plugin*

Setelah meng-*copy paste plugin*, langkah selanjutnya adalah mengaktifkan *plugin* di aplikasi RPG Maker MV. Caranya adalah klik kembali menu *Plugins Manager* dan klik baris kosong paling bawah, maka akan muncul tampilan seperti ini.



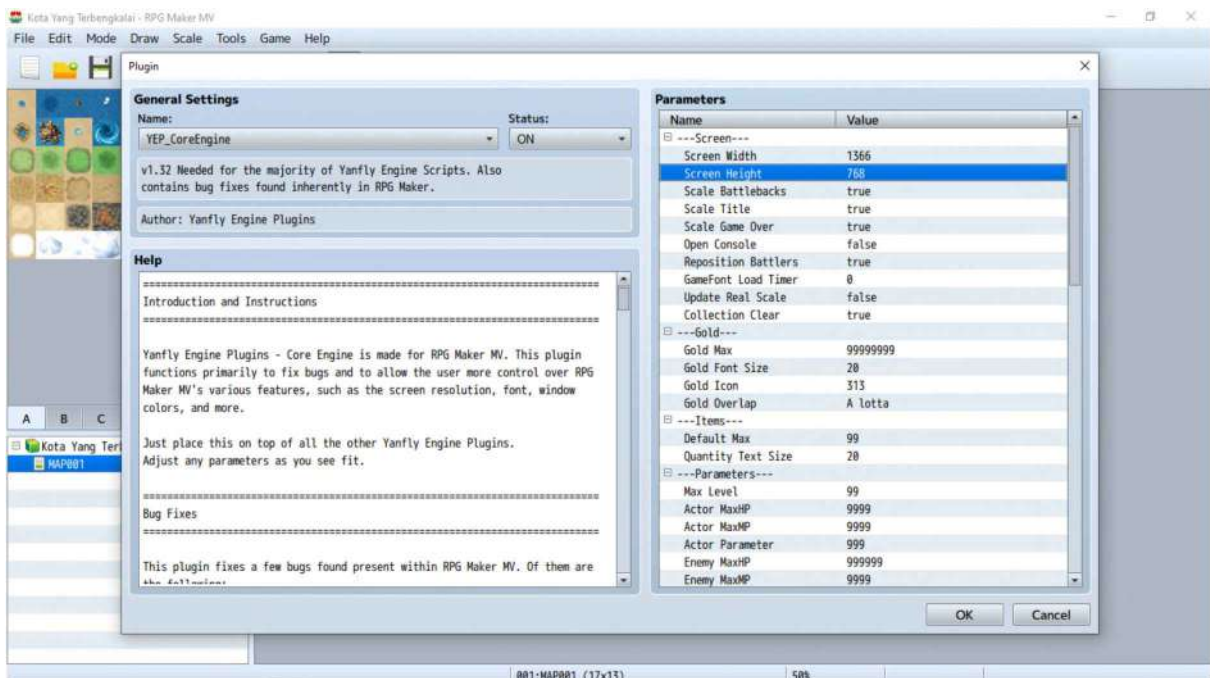
Gambar 3.9. Tampilan menambahkan *plugins*

Kemudian tambahkan satu per satu daftar *plugin* yang telah di *copy* tadi kemudian buat urutannya seperti di bawah ini.



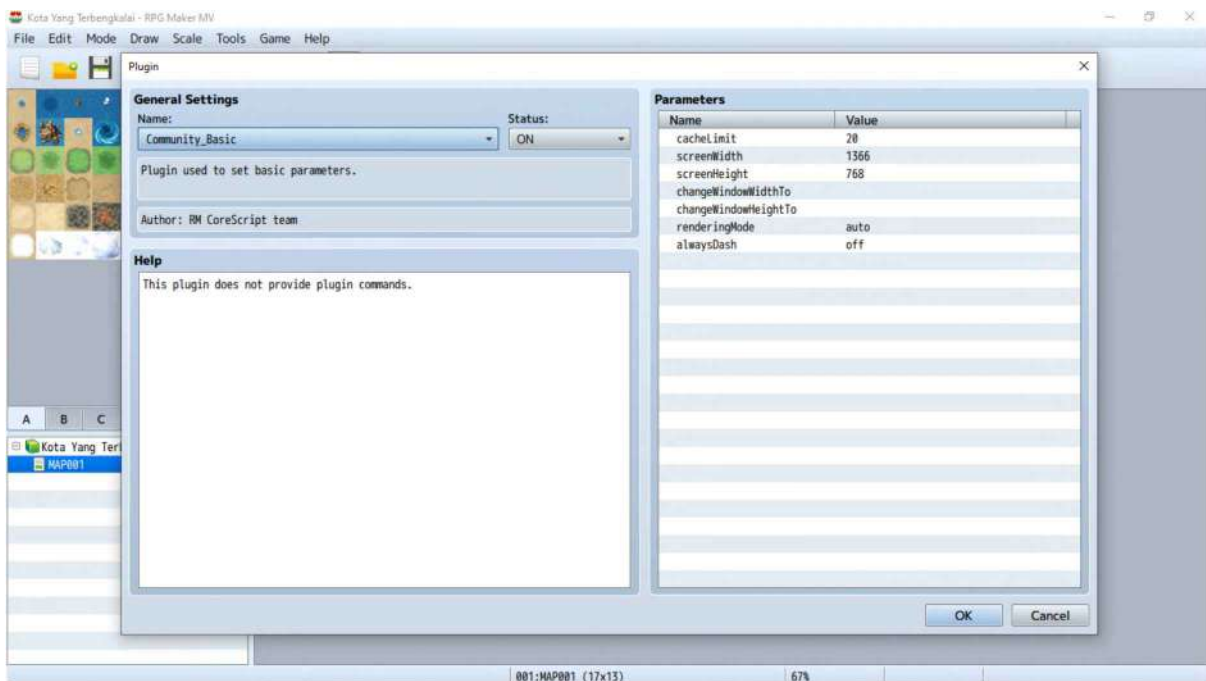
Gambar 3.10. Tampilan setelah menambahkan *plugins*

Setelah itu buka *plugin YEP_CoreEngine* dan ubah *screen width*-nya menjadi 1366 dan *screen height*-nya menjadi 768 dan klik OK.



Gambar 3.11. Tampilan *plugin YEP_CoreEngine*

Selanjutnya buka *plugin Community_Basic* dan ubah *screenWidth*-nya menjadi 1366 dan *screenHeight*-nya menjadi 768 dan klik OK. Biarkan pengaturan *plugin* lainnya secara *default*.



Gambar 3.12. Tampilan *plugin Community_Basic*

Mengatur Database

Buka menu *Database* di menu *Tools* → *Database* atau menekan tombol F9.

Actors

Setelah itu lakukan perubahan di menu *Actor* seperti di bawah ini.

1. Harold → Kenta. Ganti penampilannya seperti ini



Gambar 3.13. Menu aktor 1

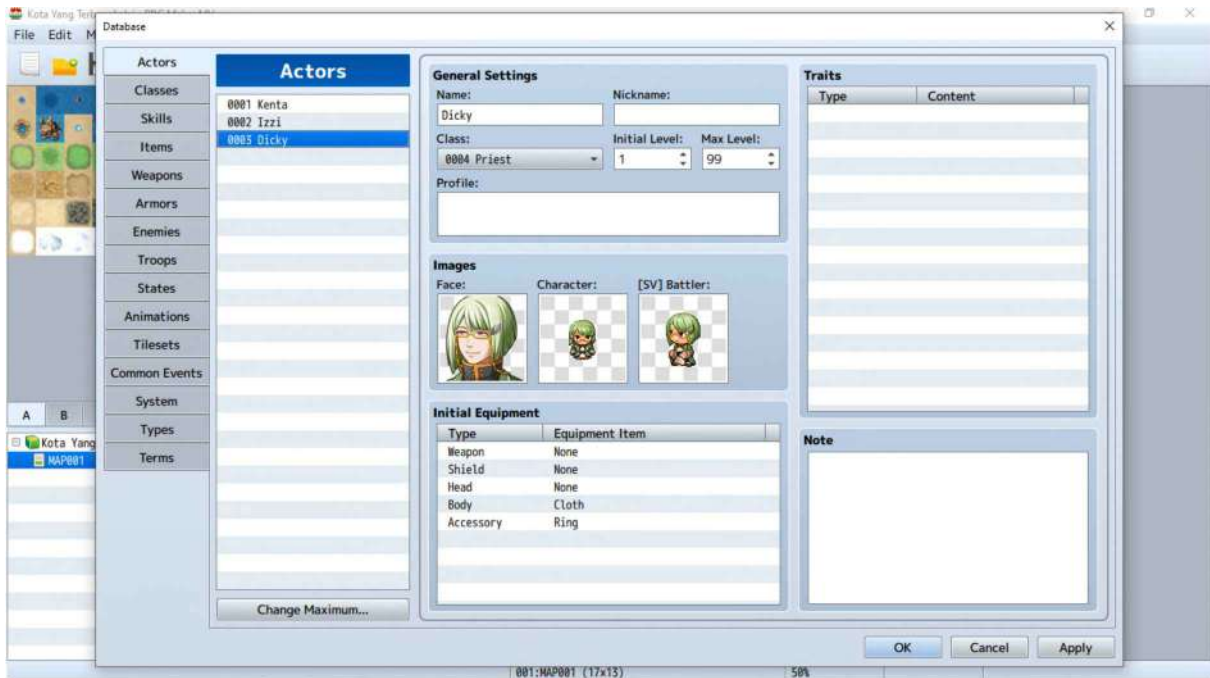
2. Hapus aktor 2 (Theresa) dengan klik kanan lalu *clear*.

3. *Copy* aktor 3 dan *paste* ke *actor 2* untuk menaikkan urutannya. Kemudian ubah seperti ini.



Gambar 3.14. Menu aktor 2

4. *Copy* aktor 4 dan *paste* ke aktor 3 untuk menaikkan urutannya. Kemudian ubah seperti ini.



Gambar 3.15. Menu aktor 3

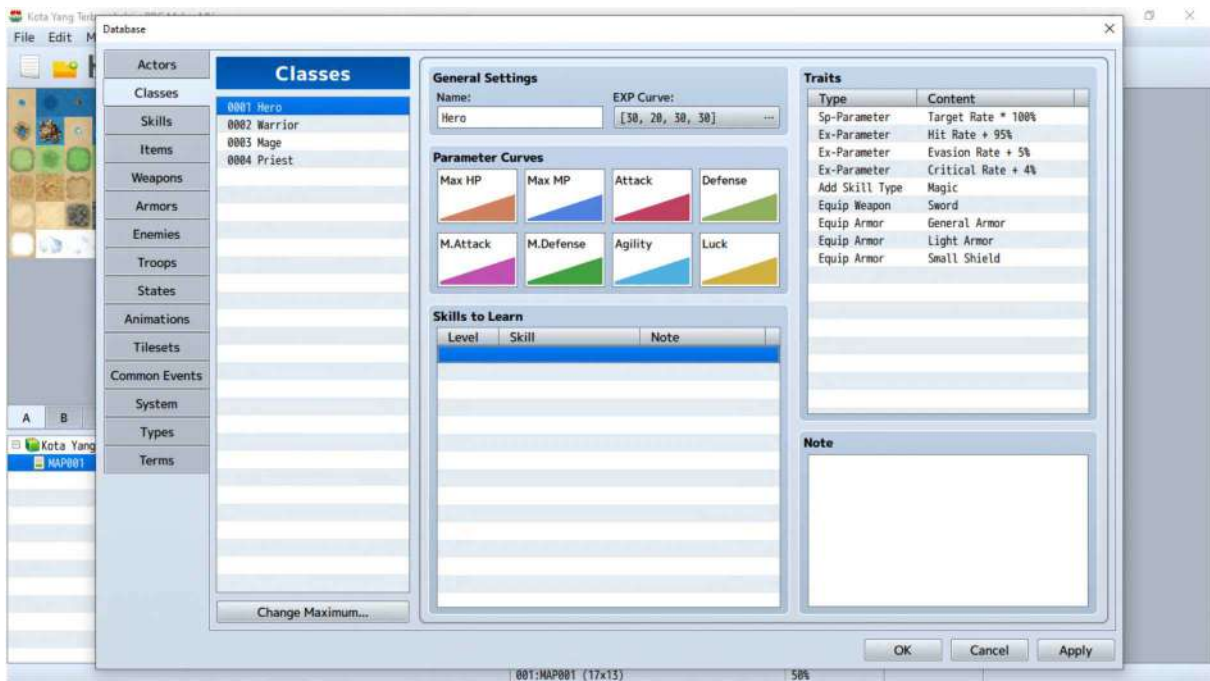
5. Klik menu *Change Maximum* dan ubah menjadi 3.



Gambar 3.16. Mengubah jumlah maksimal aktor

Classes

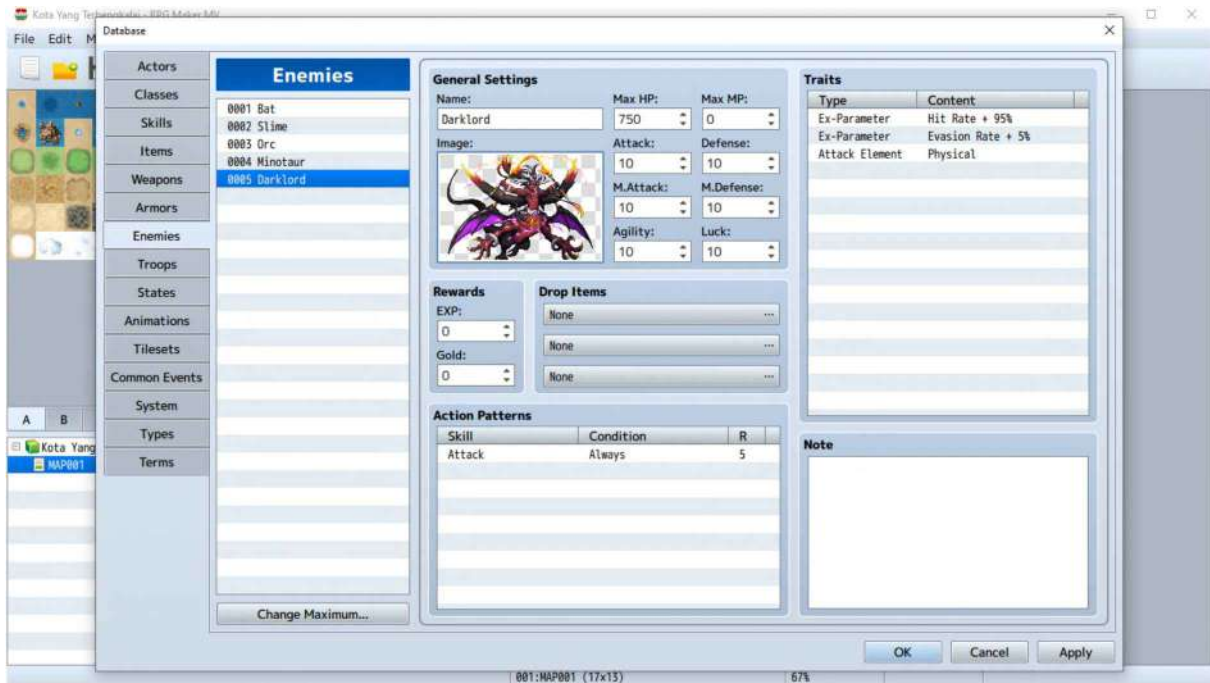
Setelah itu pilih menu selanjutnya, yaitu *Classes*. Kemudian hapus *Skill to Learn* untuk *Class Hero*, *Mage*, dan *Priest*.



Gambar 3.17. Tampilan menu *Classes*

Enemies

Selanjutnya pilih menu *Enemies* lalu klik menu *Change Maximum* dan ubah menjadi 5. Kemudian klik *Enemies* 0005 dan beri nama **Darklord**. Ubah *Max HP*-nya menjadi 750 lalu klik kotak *Image* dan pilih gambar Darklord.



Gambar 3.18. Tampilan menu *Enemies*

Troops

Setelah itu pilih menu *Troops* lalu klik menu *Change Maximum* dan ubah menjadi 5. Beri namanya menjadi **Darklord**. Kemudian klik 0005 Darklord di sebelah kanan lalu klik <Add.



Gambar 3.19. Tampilan menu *Troops*

Tilesets

Selanjutnya pilih menu *Tilesets* dan klik *Tileset 0003 Inside*. Kemudian di kolom *Images* baris Ditambahkan *SF_Inside* lalu atur seperti gambar di bawah ini.



Gambar 3.20. Tampilan *Tilesets*

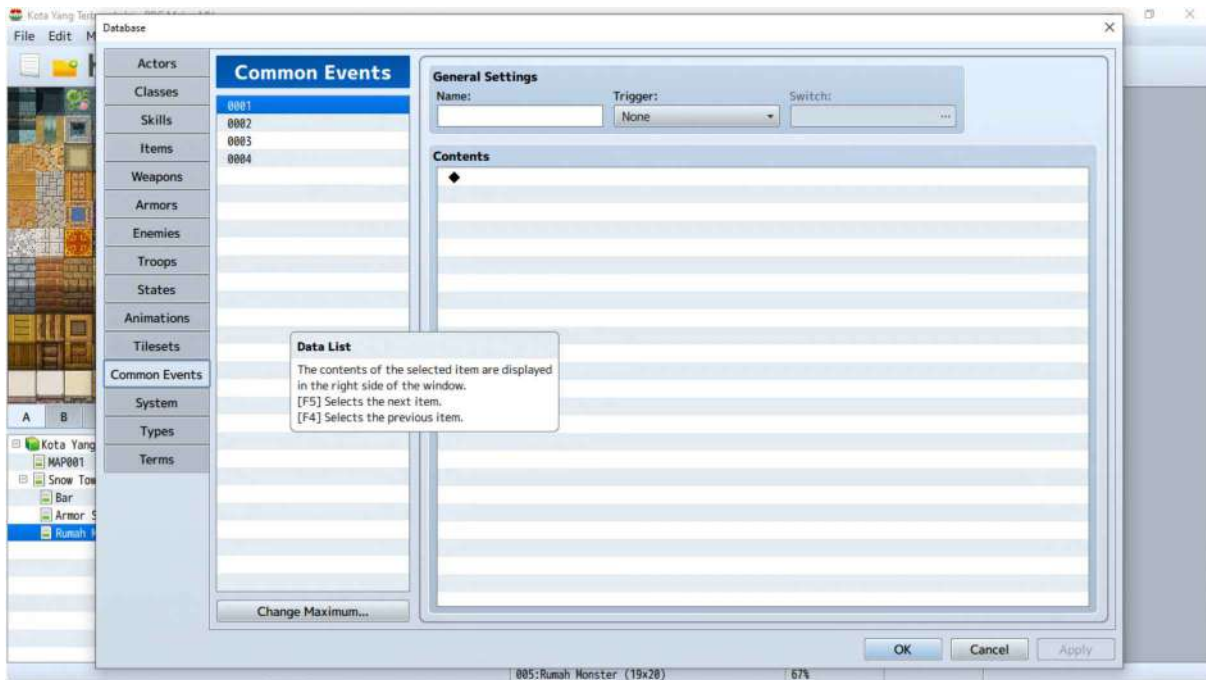


Gambar 3.21. Tampilan *Tilesets 2*

Common Events

Common Events digunakan untuk membuat *event* yang akan digunakan berulang. *Common Event* biasanya berisi *event*

yang sedikit lebih panjang sehingga menuliskannya di *Common Events* akan menghemat tempat di kotak *Contents* suatu *event*.

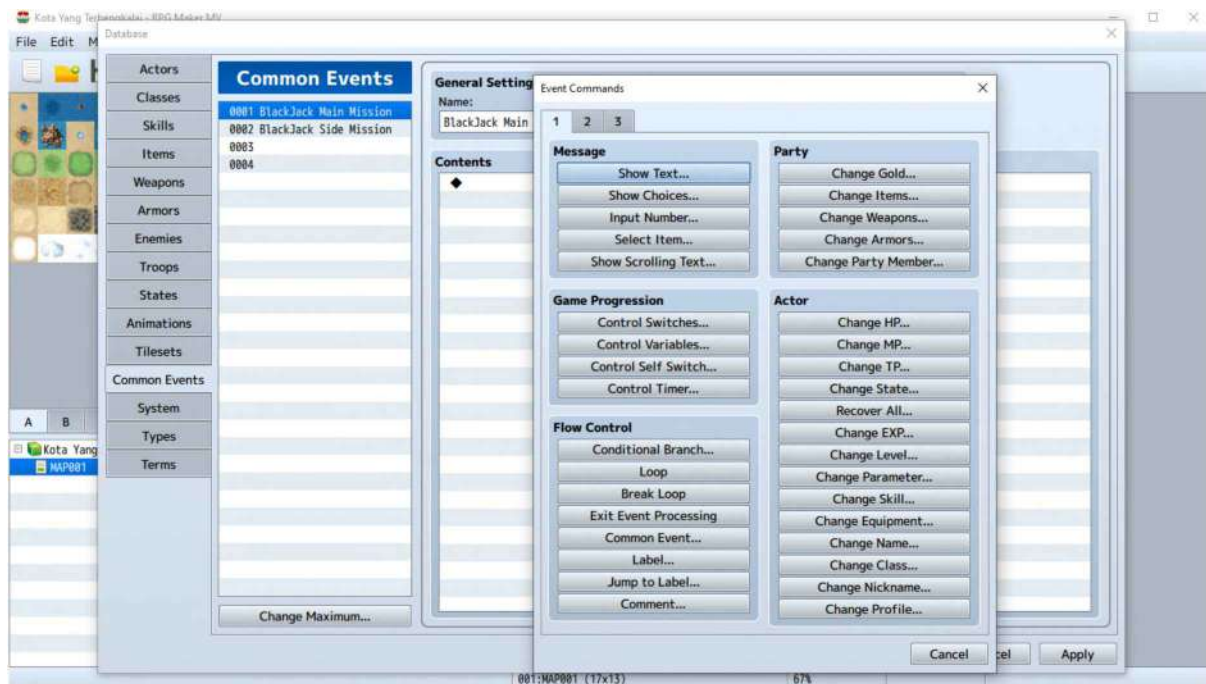


Gambar 3.22. Tampilan menu *Common Events*

Setelah muncul gambar seperti di atas lalu klik *Common Events* 0001 dan beri nama ***BlackJack Main Mission*** lalu klik *Common Events* 0002 dan beri nama ***BlackJack Side Mission***.

BlackJack adalah salah satu permainan kartu yang mana pemain harus menjumlahkan kartunya sampai 21 untuk menang atau memiliki sejumlah 5 kartu yang nilainya ≤ 21 . Jika jumlah kartu pemain melebihi 21 maka pemain akan kalah. Pembuatan *event* ini saya ambil dari YouTube (Haydengames3v2, 2016) yang sedikit dimodifikasi.

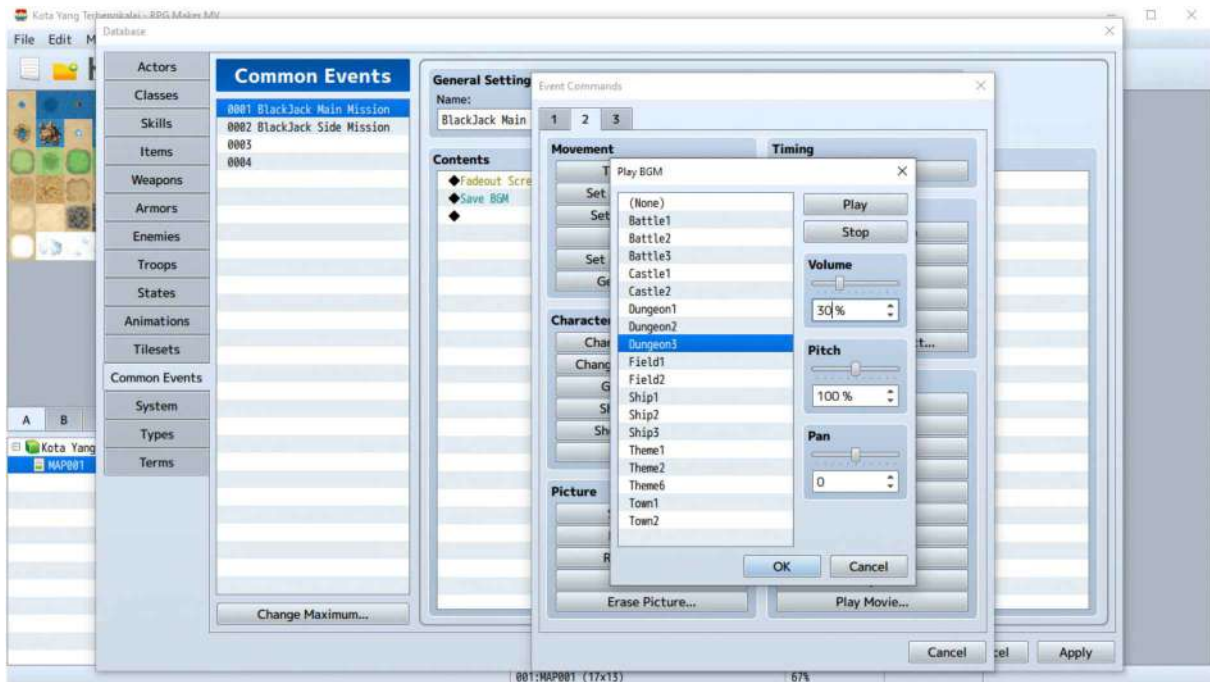
Selanjutnya klik 0001 *BlackJack Main Mission* kemudian klik dua kali di baris *Contents* maka akan muncul seperti ini.



Gambar 3.23. Tampilan *Event Commands*

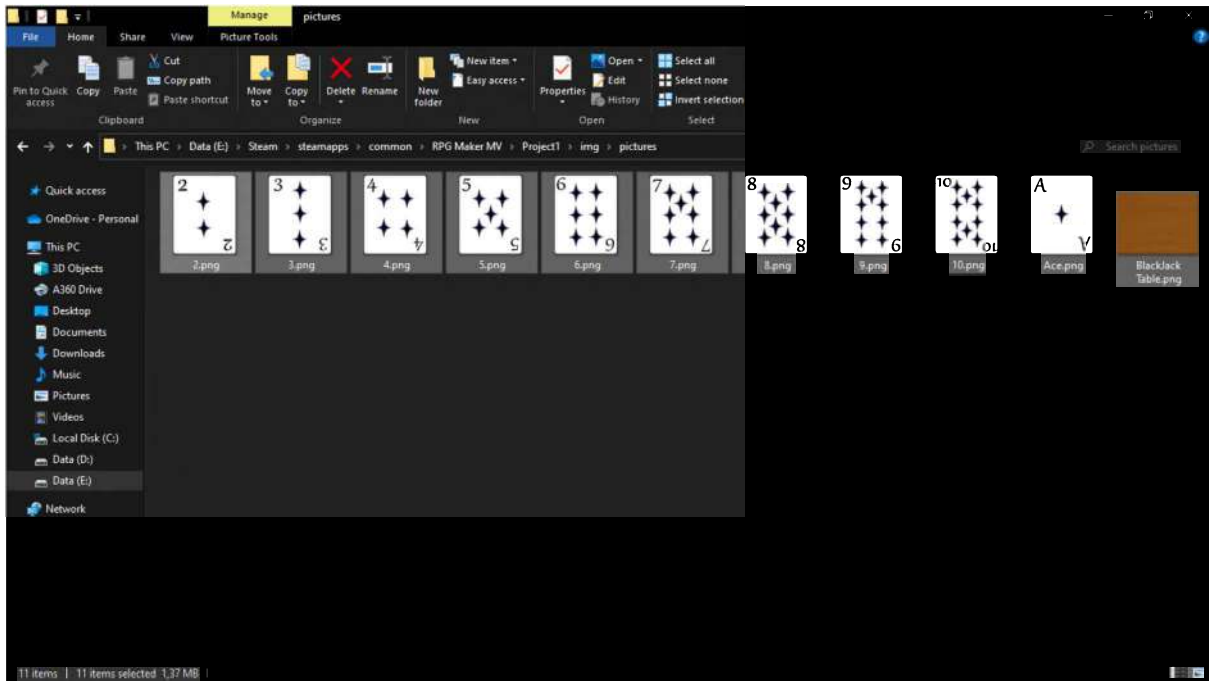
Kemudian klik halaman dua lalu pilih *Fadeout Screen* di kolom *Screen*. *Event Fadeout Screen* akan melakukan transisi layar menjadi hitam.

Setelah itu klik baris di bawah *Fadeout Screen* dan tambahkan *event Save BGM* di kolom *Audio & Video*. Selanjutnya tambahkan *event Play BGM* lalu pilih *Dungeon3* dan atur volumenya 30%. Saya menyarankan untuk volume diatur di antara 25–40% agar suara tidak terlalu keras saat *game* dimainkan.



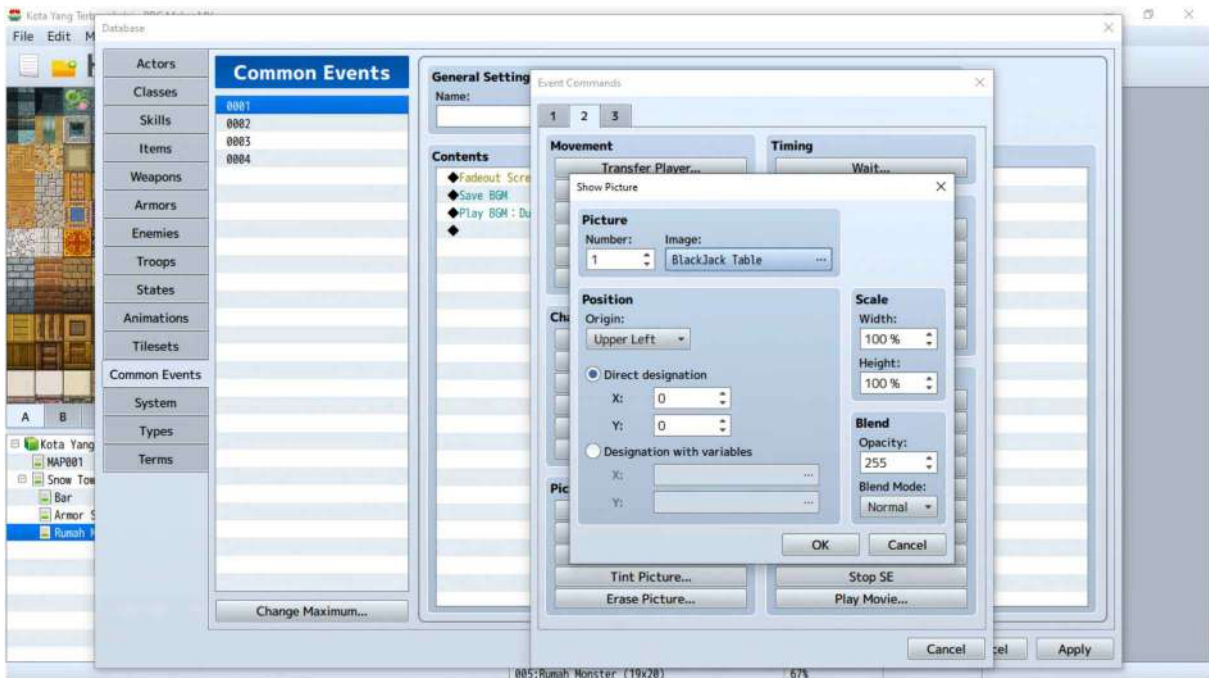
Gambar 3.24. Tampilan *Event Play BGM*

Sebelum membuat *event* berikutnya, impor gambar terlebih dulu yang telah saya sediakan di <https://s.id/KotaYangTerbengkalai> kemudian letakkan di folder dokumen *game* anda → *img* → *pictures*.



Gambar 3.25. Mengimpor gambar ke RPG Maker MV

Selanjutnya baris di bawah *Play BGM* dan pilih *Show Picture* kemudian atur seperti gambar di bawah ini lalu klik OK.

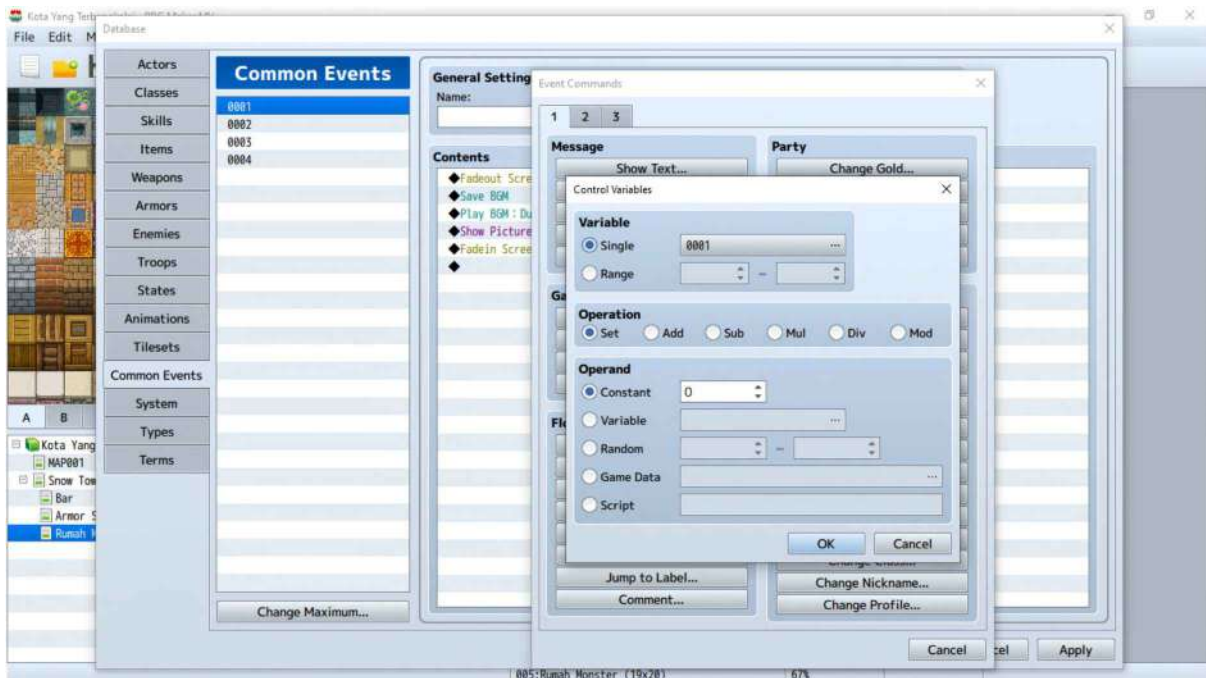


Gambar 3.26. Membuat event *Show Picture*

Setelah itu klik baris di bawah *Show Picture* dan pilih *Fadein Screen* untuk transisi layar dari hitam ke tampilan semula.

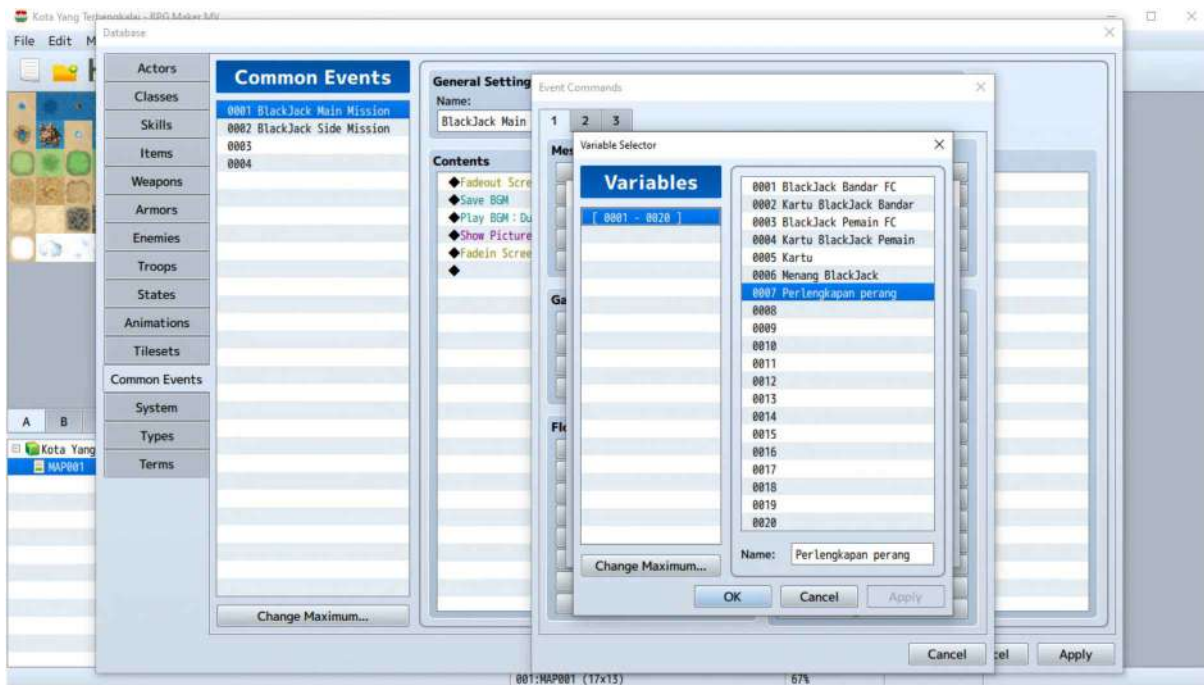
Membuat Variabel

Langkah selanjutnya adalah membuat variabel menggunakan *event Control Variable*. Klik baris di bawah *Fadein Screen* kemudian pilih *Control Variable* yang akan muncul jendela seperti ini.



Gambar 3.27. Membuat *event Control Variable*

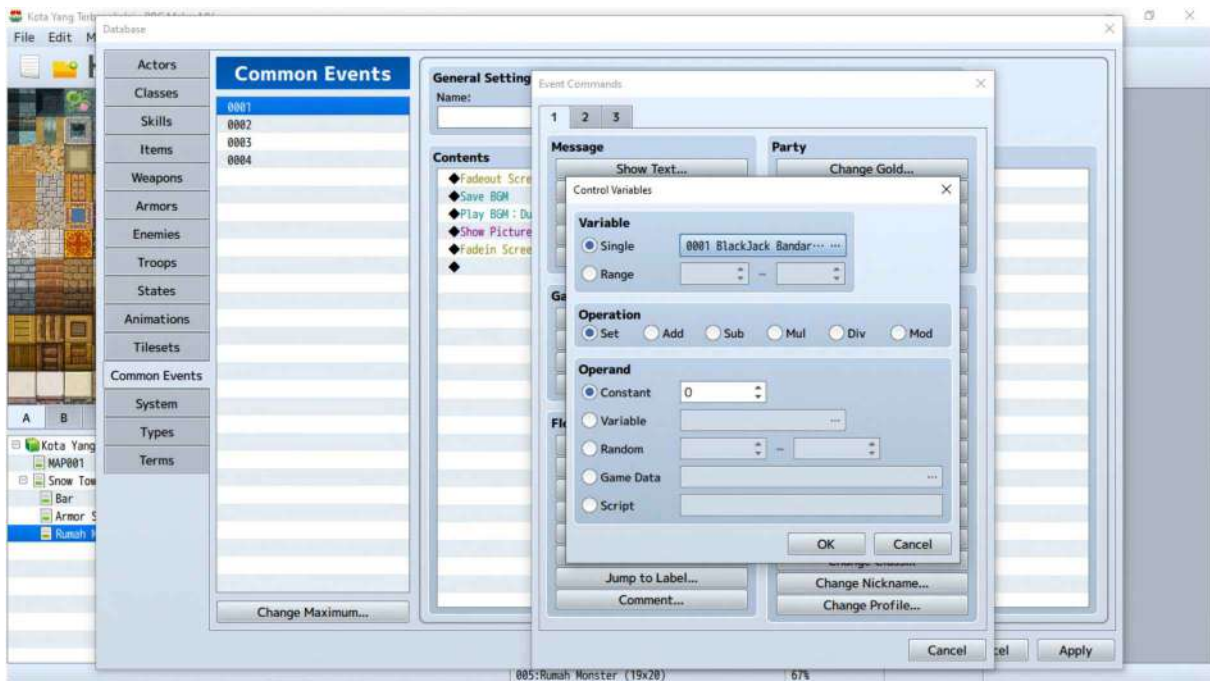
Setelah muncul tampilan seperti di atas kemudian klik kotak #0001 di sebelah kanan kolom *Single* lalu lakukan seperti ini.



Gambar 3.28. Membuat Variabel

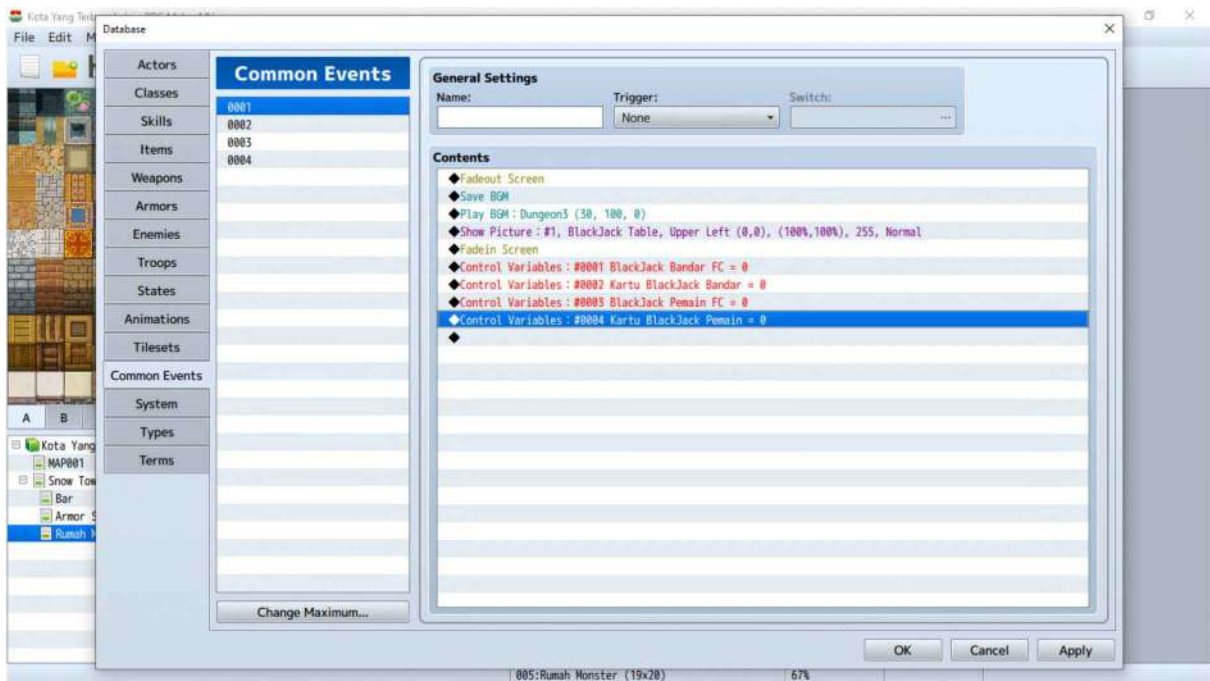
Variabel tersebut nantinya akan digunakan pada *event* di *game* ini. Sebagai contoh untuk variabel nomor satu, FC pada *BlackJack* Bandar FC memiliki kepanjangan *First Card* yang berarti kartu yang didapatkan Bandar dari putaran pertama sama artinya dengan variabel nomor tiga. Variabel nomor dua berarti jumlah angka kartu yang dimiliki oleh Bandar.

Setelah itu pilih variabel nomor satu, yaitu variabel *BlackJack* Bandar FC kemudian klik *Apply* lalu *OK*.



Gambar 3.29. Membuat Variabel BlackJack Bandar FC

Kemudian lakukan seperti gambar di atas kemudian klik OK. *Control Variable* kali ini akan mengatur variabel *BlackJack* Bandar FC menjadi = 0 yang akan berfungsi sebagai reset *variable BlackJack* Bandar FC saat pertama kali bermain. Lakukan hal yang sama sampai variabel 0004.



Gambar 3.30. Membuat beberapa *Control Variables*

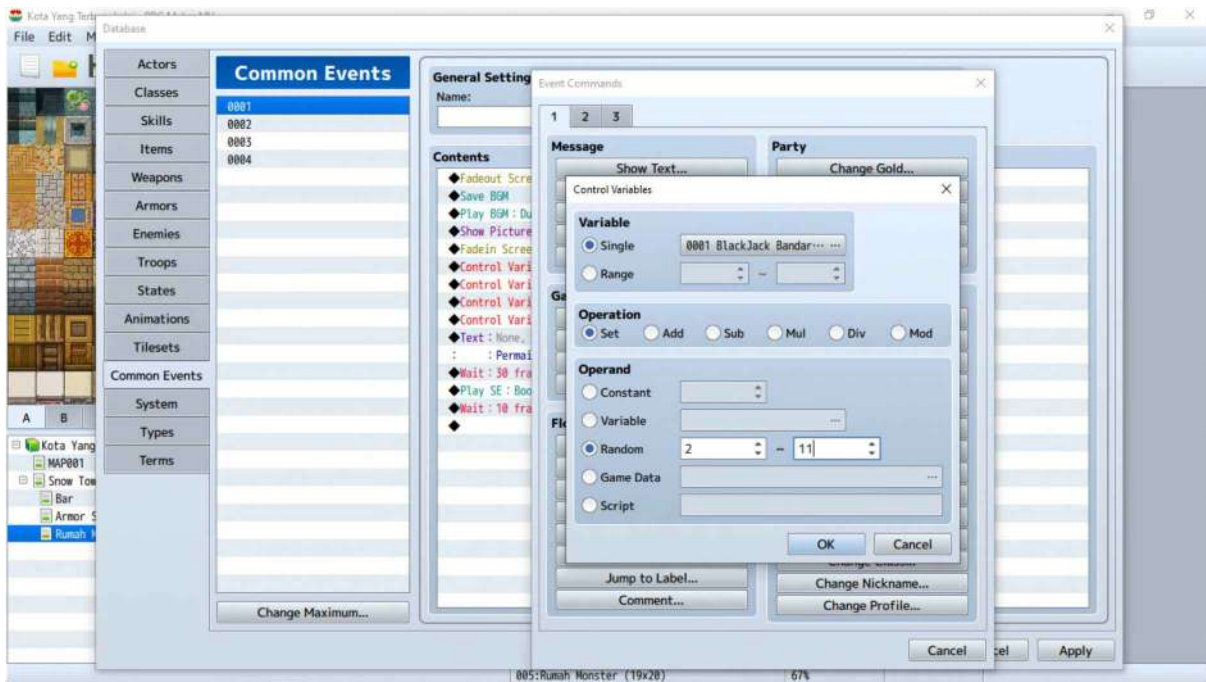
Selanjutnya tambahkan beberapa *event* dengan tabel di bawah ini.

- ◆Text : None, Window, Bottom
: Permainan dimulai!
- ◆Wait : 30 frames
- ◆Play SE : Book1 (90, 100, 0)
- ◆Wait : 10 frames

Keterangan:

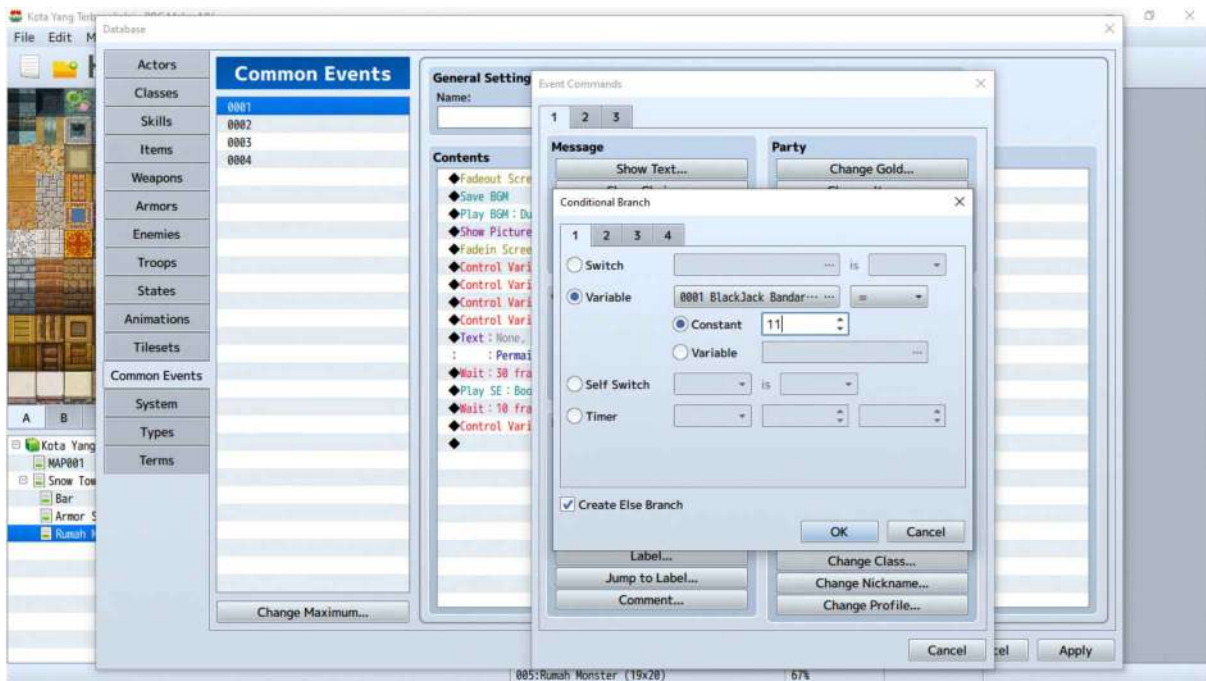
- *Wait*: Menunggu dengan durasi *frame* yang telah ditentukan sebelum *event* selanjutnya berjalan.
- *Play SE*: Memainkan *sound effect*.

Setelah itu buat *Control Variable* lagi di bawah *event Wait 10 frames*. Pilih Variabel 0001 dan atur seperti gambar di bawah ini.



Gambar 3.31. Membuat *Control Variables Random Operand*

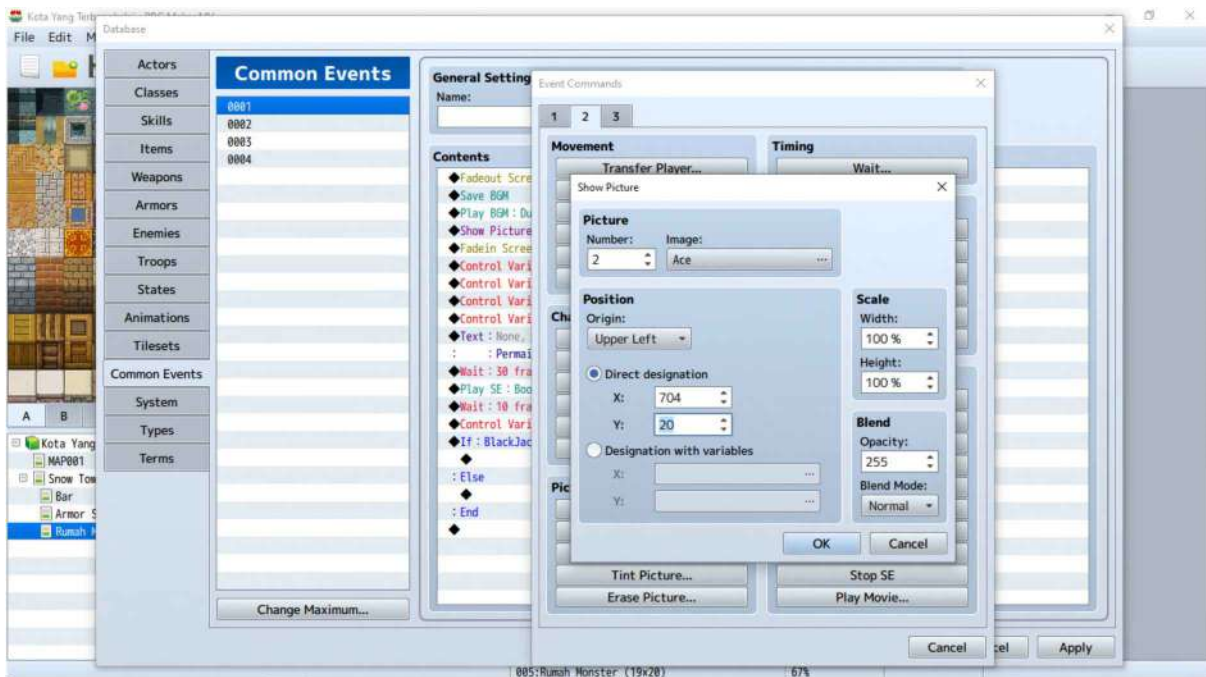
Control Variable Random Operand di atas adalah untuk memunculkan kemungkinan kartu yang keluar, yaitu kartu 2-10 dan 11 yang berarti kartu AS. Selanjutnya tambahkan *event Conditional Breach* dan atur seperti gambar di bawah ini lalu klik OK.



Gambar 3.32. Membuat *Conditional Branch BlackJack*

Conditional Branch adalah *event* yang akan berjalan pada kondisi tertentu sesuai namanya,. Pada gambar di atas adalah pembuatan *Conditional Branch* untuk kondisi *If: BlackJack Bandar FC = 11*. Jika kondisi tersebut tidak terpenuhi maka *event* di bawahnya tidak akan berjalan.

Setelah membuat *Conditional Branch*, tambahkan *event Show Picture* di bawah baris *If: BlackJack Bandar FC = 11* untuk menambahkan gambar kartu dan atur seperti ini.



Gambar 3.33. Menambahkan gambar kartu

Setelah menambah *event Show Picture* kemudian masukkan *event Show Text* di bawah ini.

◆Text : None, Window, Bottom

: Bandar: AS!

Kemudian di bawah baris *Else* tambahkan *Conditional Branch* lagi dan jangan setiap lupa membuat *Conditional Branch* centang bagian *Create Else Branch*.

Karena *Common Event* terlalu panjang untuk ditulis, maka akan saya menyediakan *file* khusus untuk *Common Event BlackJack Main Mission* dan *BlackJack Side Mission* yang bisa anda *download* di <https://s.id/KotaYangTerbengkalai> dan bisa anda salin *event code*-nya.

System

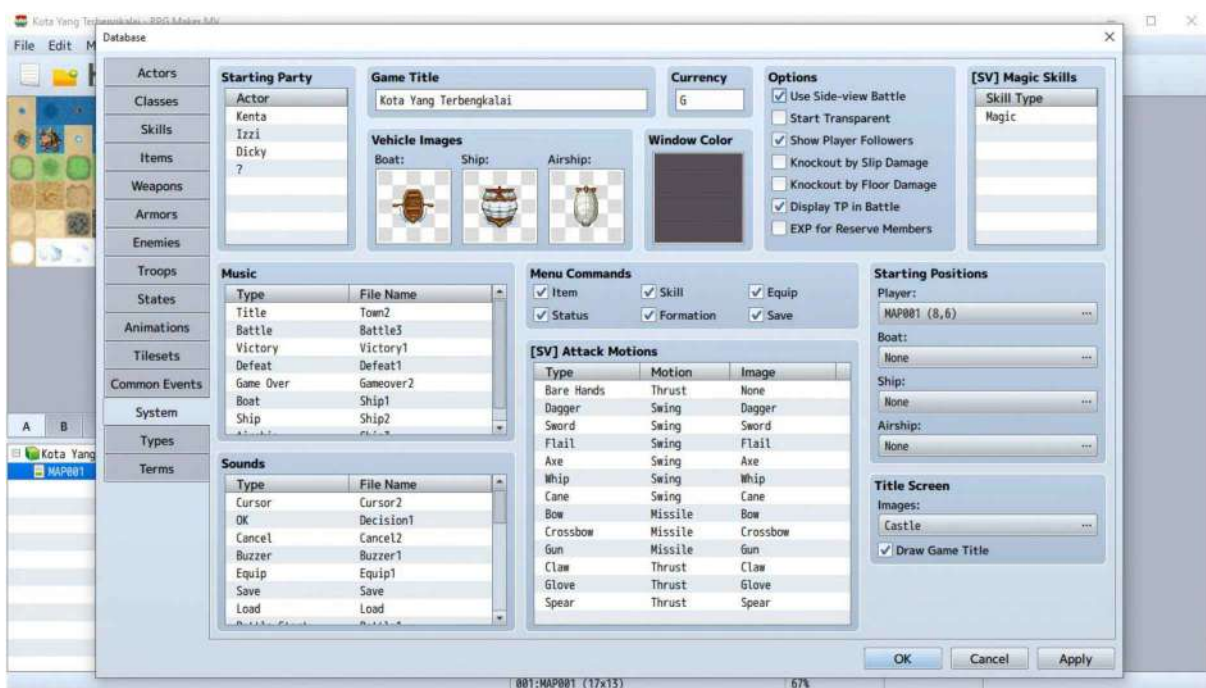
Selanjutnya pilih menu *System* dan pilih kolom *Music* lalu ganti menjadi seperti ini.

Title → Town2

Battle → Battle3

Game Over → Gameover2

Kemudian ubah volumenya menjadi 25%. Kemudian centang pada tulisan *Use Side-view Battle* di kolom *Option* sebelah kanan.

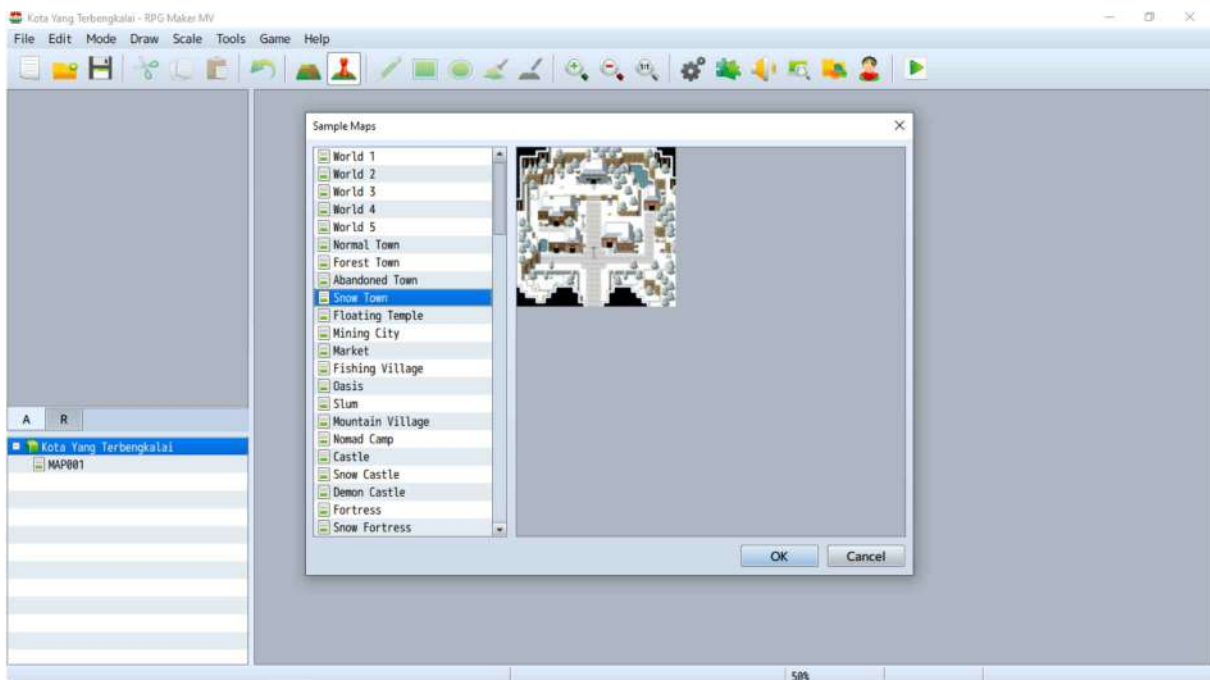


Gambar 3.34. Tampilan menu *System*

Membuat Map

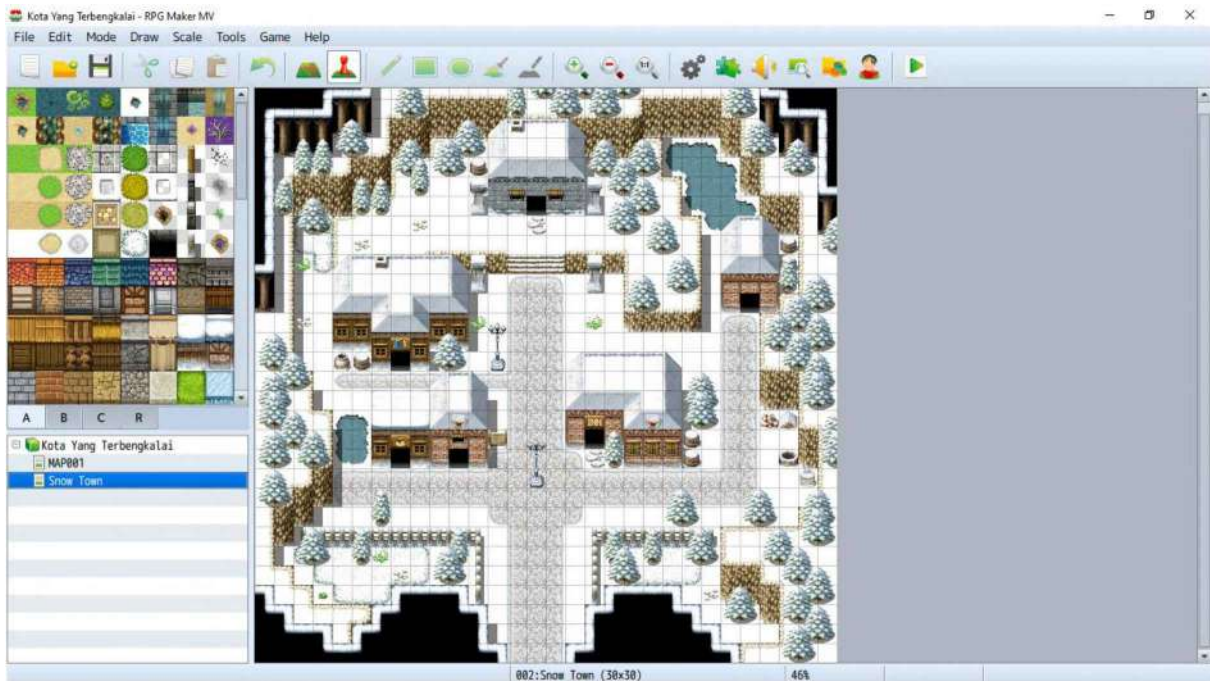
Snow Town

Klik kanan pada menu yang tertulis judul *game*, kemudian pilih *Load*. Setelah itu pilih *map Snow Town* lalu OK.



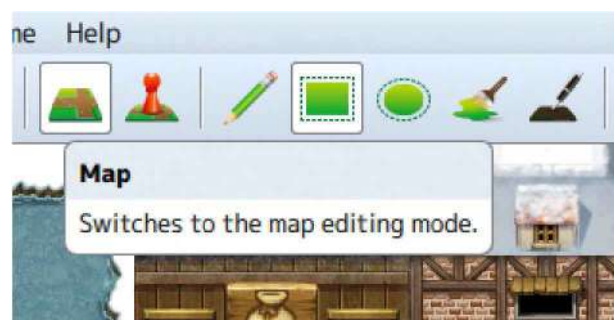
Gambar 3.35. Tampilan *Sample Map*

Jika sudah maka tampilannya akan seperti ini.



Gambar 3.36. Tampilan *map Snow Town*

Lakukan sedikit modifikasi pada untuk *map* ini dengan menambahkan objek *Barricade (H, Snow)* di koordinat (12,25) dan (16,25). Cara menambah atau mengedit objek bisa dengan klik tombol bernama *map* di bagian atas.



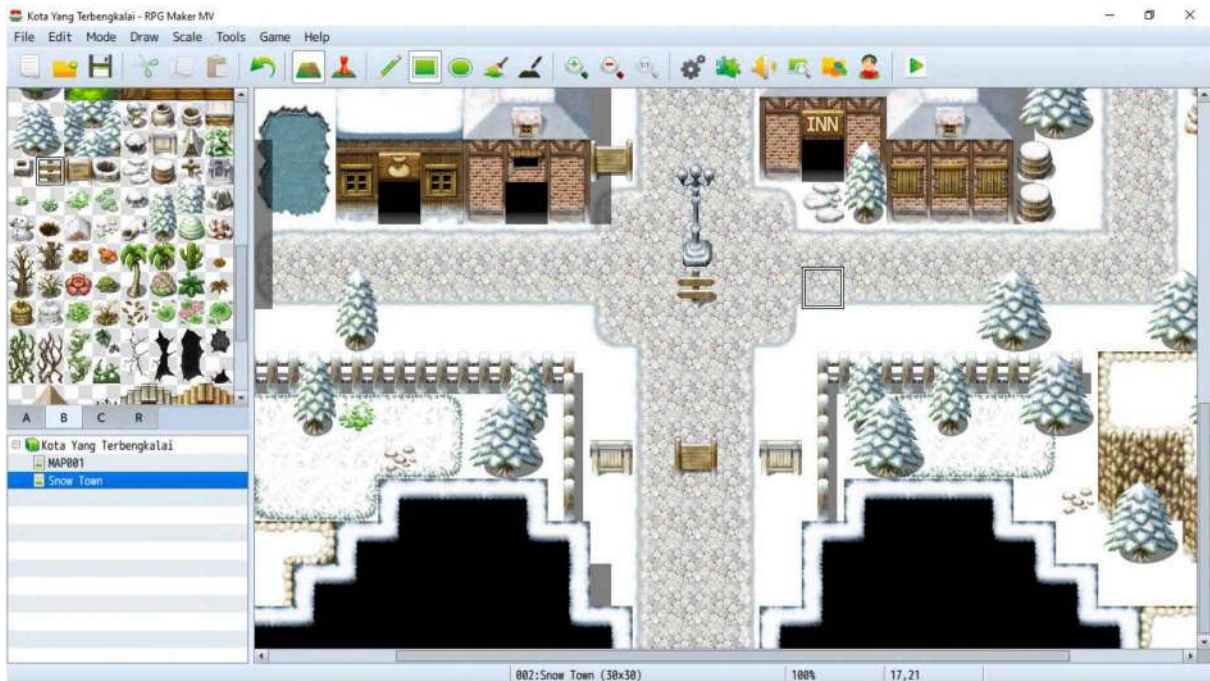
Gambar 3.37. Tombol *Map* untuk mengedit objek

Objek *barricade* terletak di *tilesets B*.



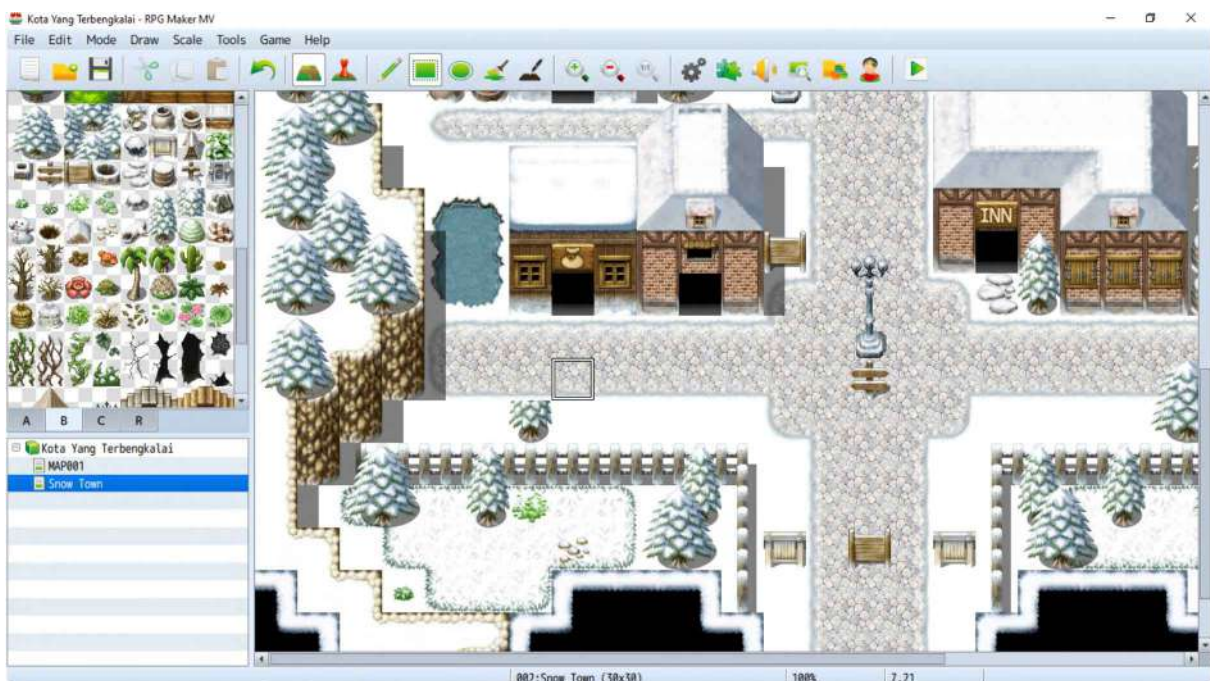
Gambar 3.38 *Tileset* letak objek *Barricade (H, Snow)*

Setelah menambahkan objek *Barricade (H, Snow)*, objek selanjutnya adalah *Sign A (Snow)* di koordinat (14,21) dan *Sign B (Snow)* di koordinat (14,25). Maka hasilnya akan seperti ini.



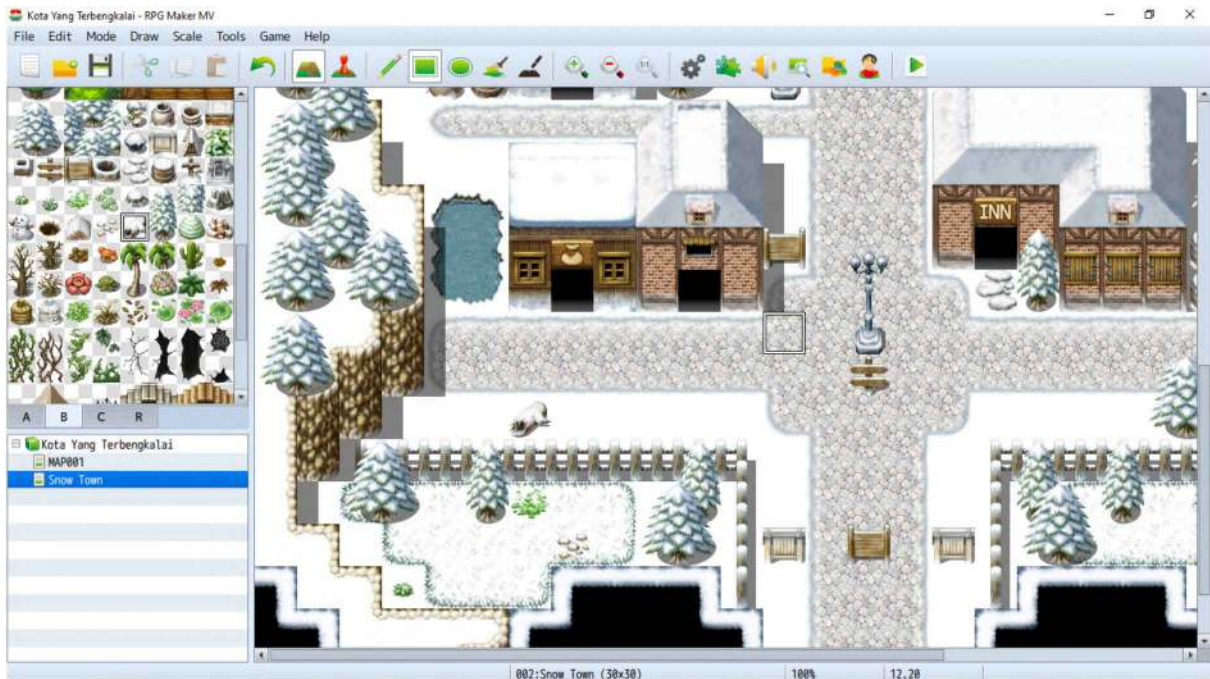
Gambar 3.39. Tampilan *map* setelah menambahkan beberapa objek

Selanjutnya ubah pohon yang berada di koordinat (6,21-22) menjadi batang pohon dengan cara klik kanan pada *tile* (piksel pada *map*) jalan lalu klik kiri ke koordinat pohon (6,21).



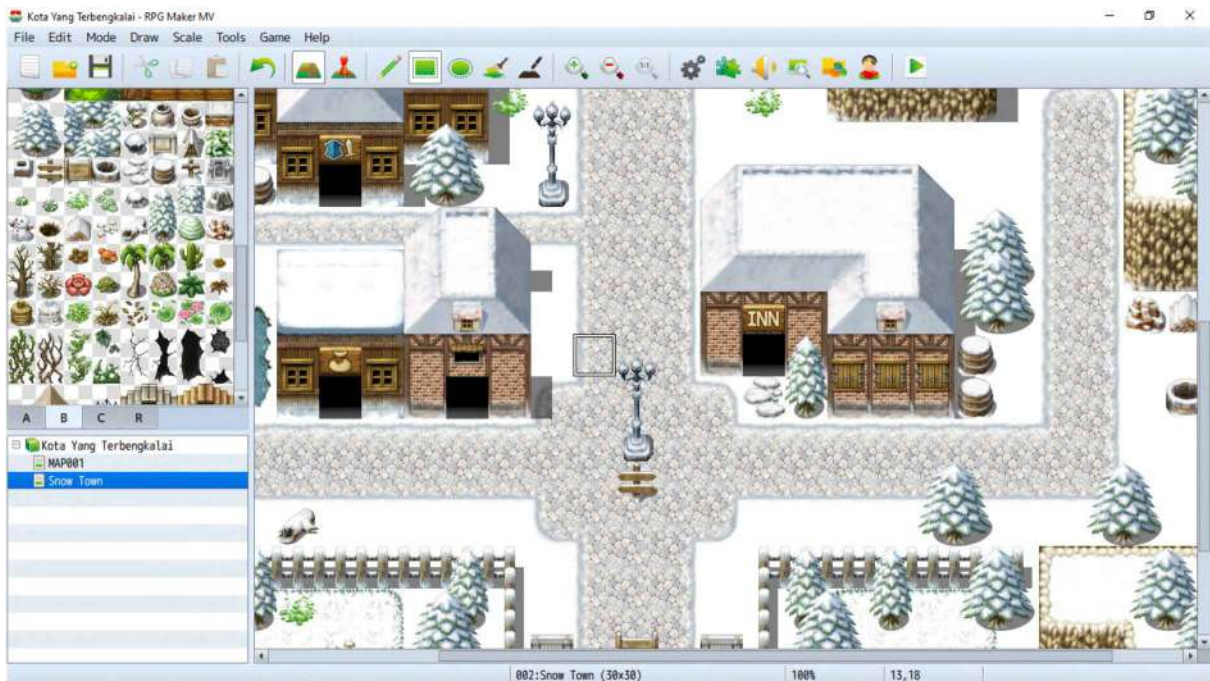
Gambar 3.40. Meng-*copy* dan *paste* objek yang ada di *map*

Setelah itu *copy tile* salju kemudian klik di koordinat (6,22) pohon agar pohon terhapus. Setelah terhapus, letakkan objek *Fallen Log (Snow)*.



Gambar 3.41. Mengubah objek

Selanjutnya hapus papan yang ada di koordinat (12,18) dengan cara yang sama seperti di atas yang akan memperoleh hasil seperti ini.



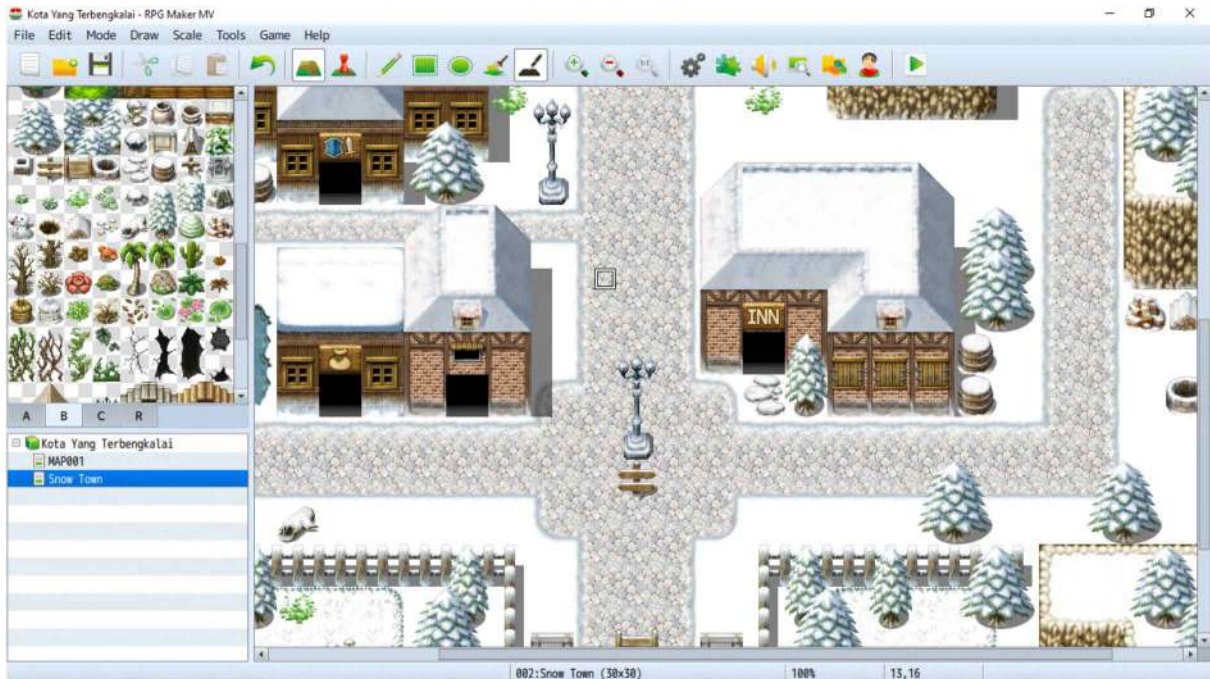
Gambar 3.42. Menghapus objek bayangan

Setelah papan terhapus, bayangan akan ikut terhapus karena posisi bayang berada di belakang objek. Cara mengembalikan bayangannya adalah dengan *menu Shadow Pen* yang ada di atas.



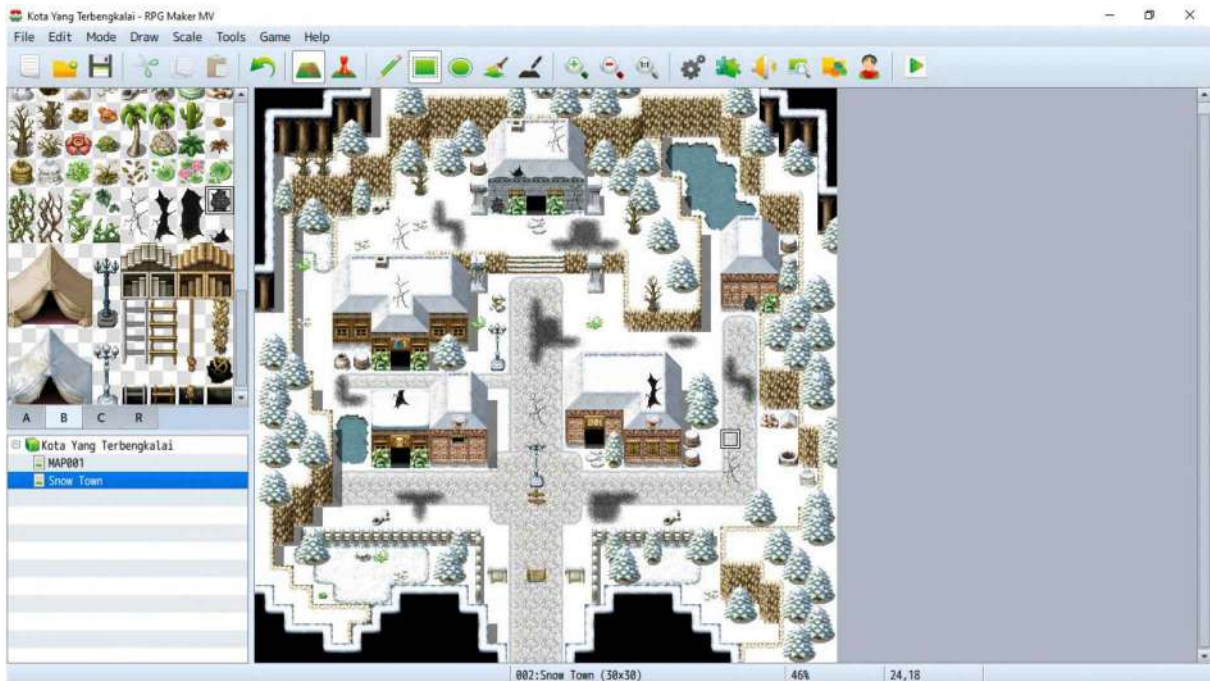
Gambar 3.43. Tombol *Shadow Pen* untuk menambahkan bayangan

Kemudian tambahkan bayangan ke koordinat papan, yaitu (12,17-18). Maka hasilnya akan seperti ini.



Gambar 3.44. Menambahkan bayangan

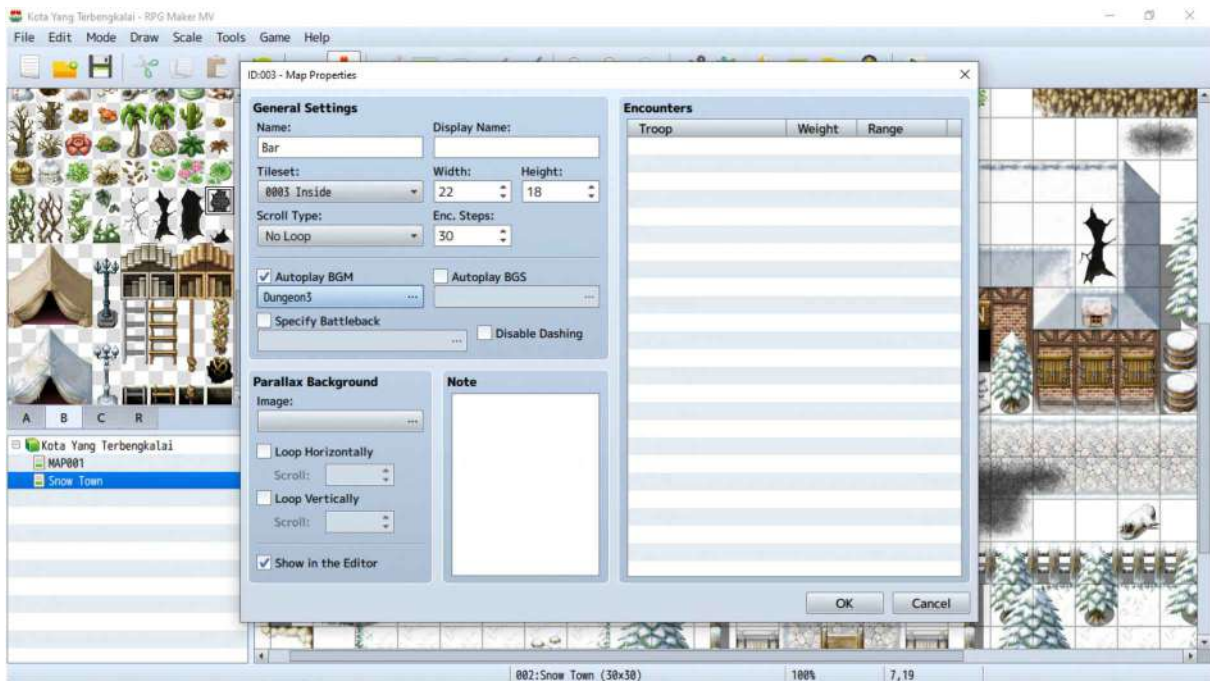
Ulangi cara di atas dengan objek yang berbeda sampai hasilnya seperti ini.



Gambar 3.45. Meng-edit *map Snow Town*

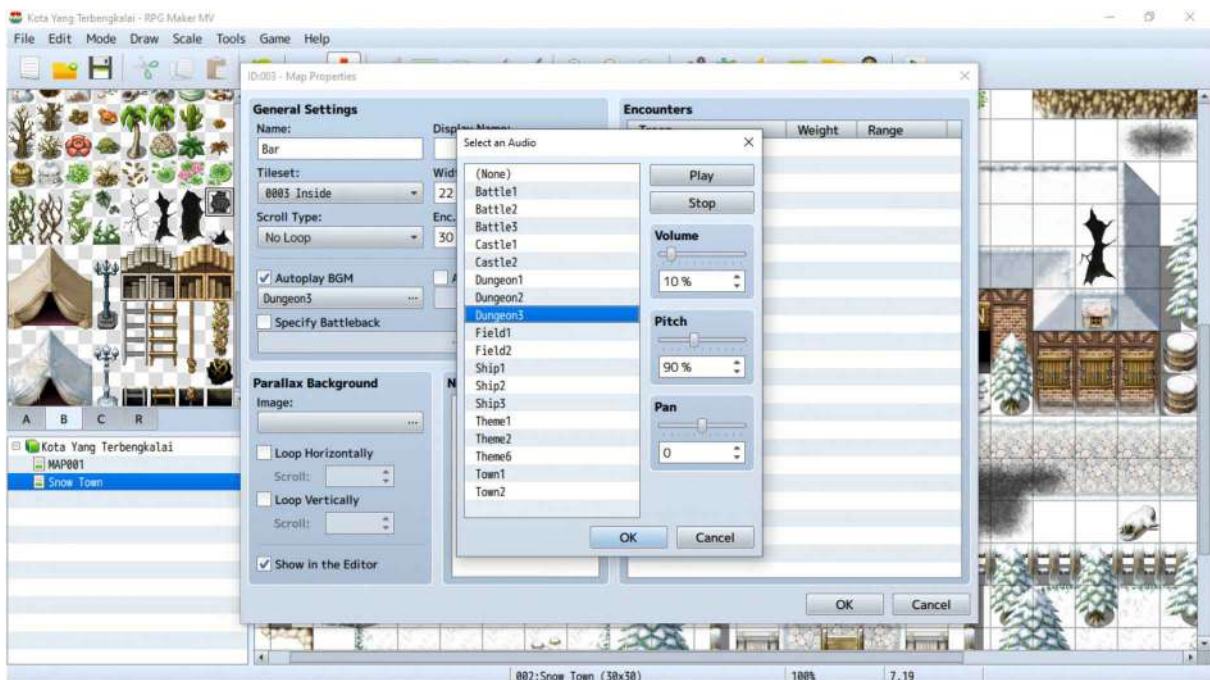
Bar

Langkah selanjutnya adalah membuat *map* baru, yaitu *map* Bar di bawah *map Snow Town*. Caranya adalah klik kanan pada *Snow Town* lalu klik *New*. Kemudian atur seperti di bawah.



Gambar 3.46. Mengatur *map* baru

Untuk *Autoplay BGM*, atur seperti ini.



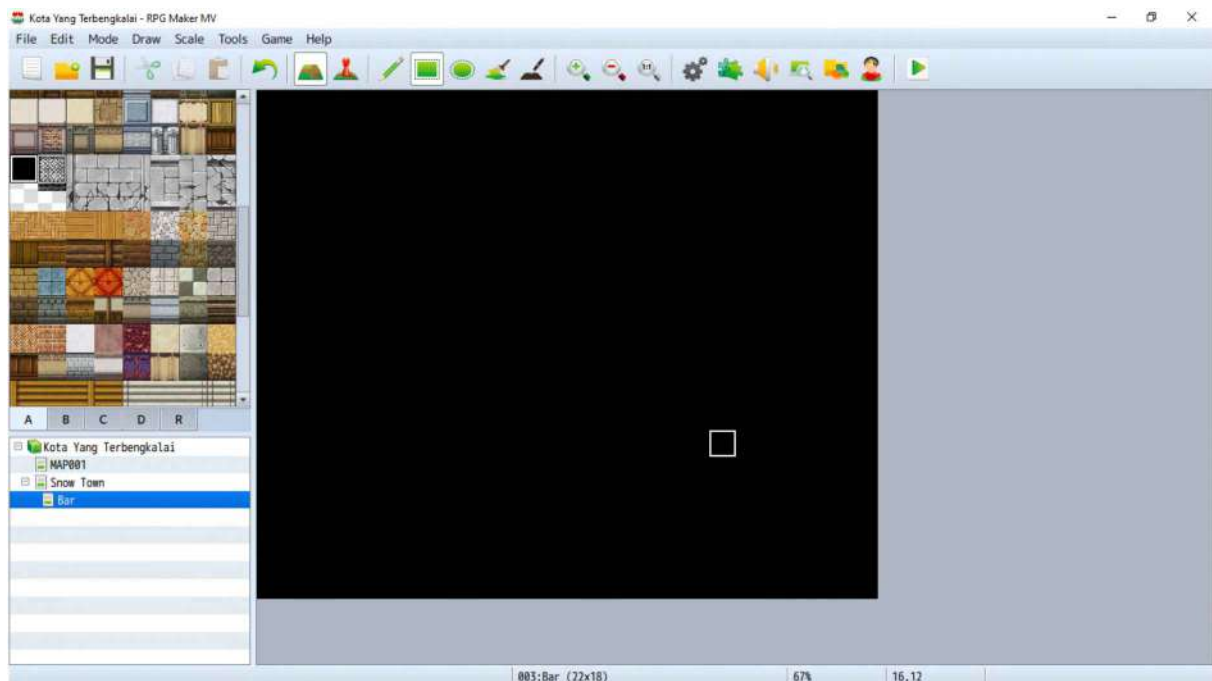
Gambar 3.47. Mengatur *BGM*

Setelah *map* terbuat, blok *map* dengan warna hitam menggunakan *tile* bernama *Darkness* yang ada di *tilesets A*.



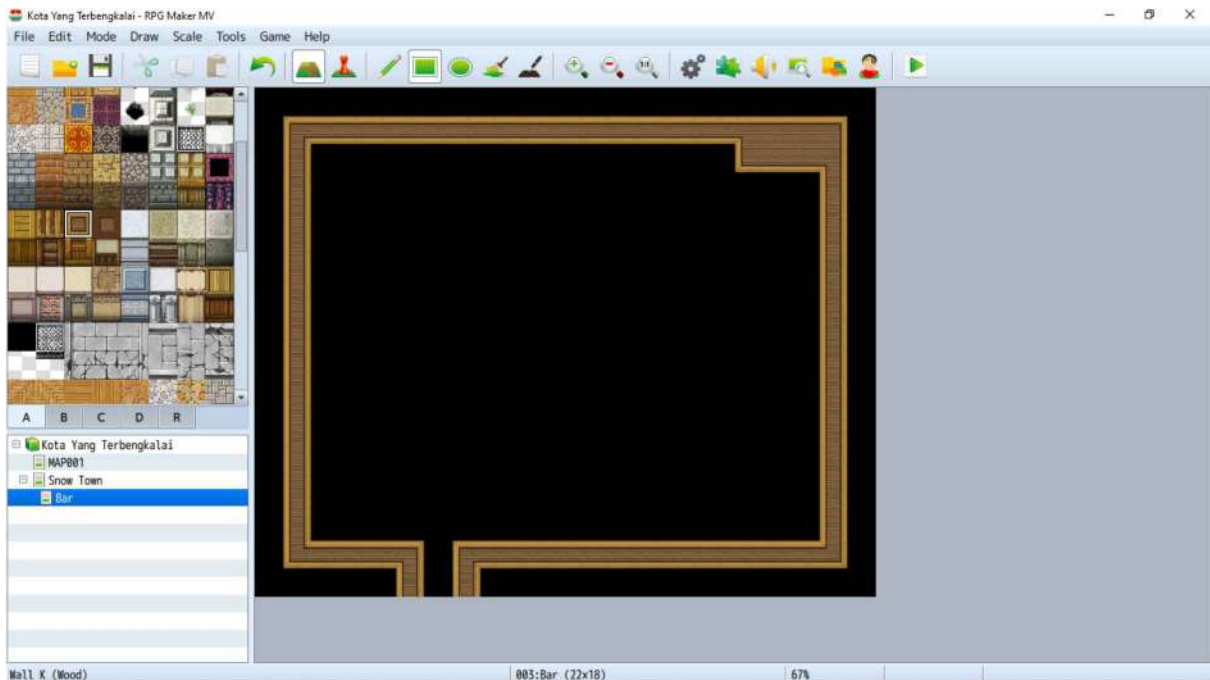
Gambar 3.48. *Tile Darkness*

Maka akan terlihat seperti ini.



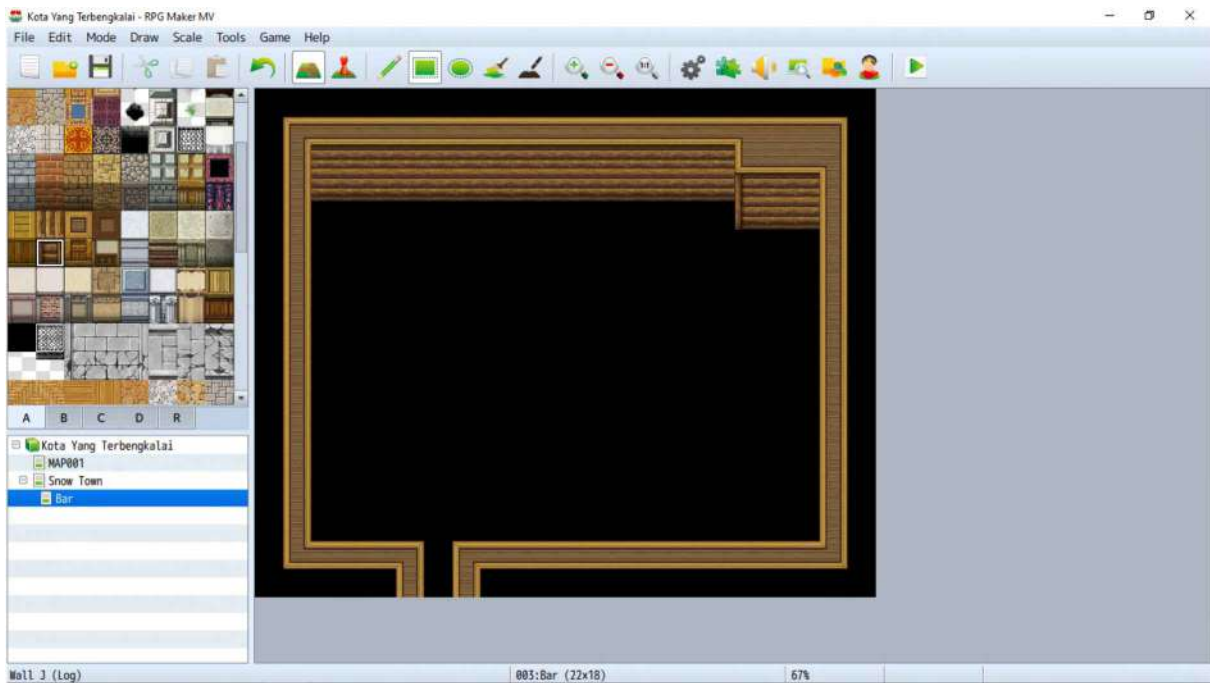
Gambar 3.49. Memblok *map*

Setelah itu pilih *tile Wall K (Wood)* di *tilesets A* dan buat seperti ini.



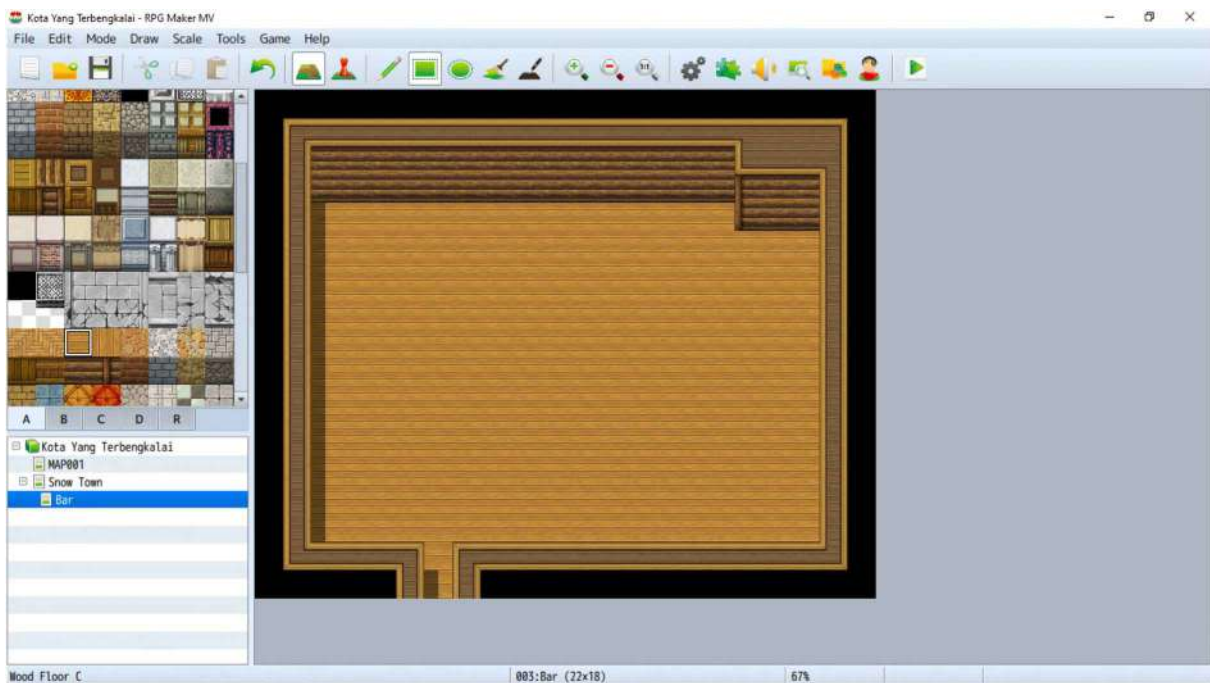
Gambar 3.50. Membuat *outline map Bar*

Kemudian pilih *tile Wall J (Log)* dan buat seperti ini.



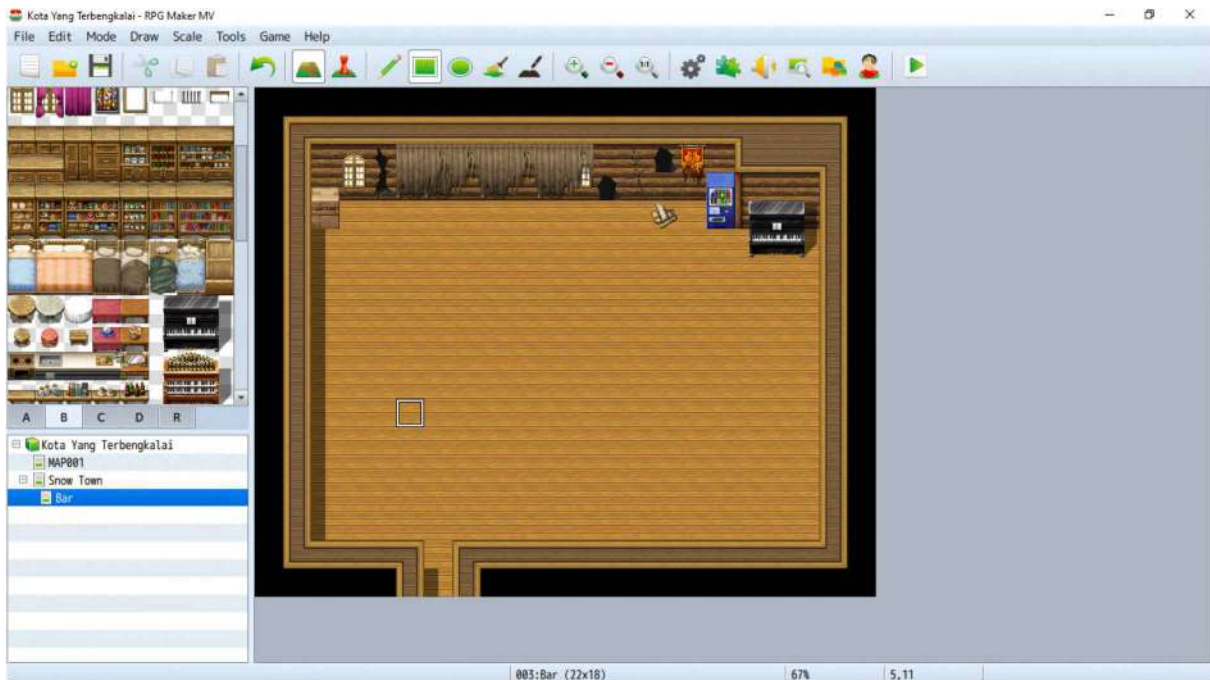
Gambar 3.51. Membuat dinding *map Bar*

Setelah itu pilih *tile Wood Floor C* dan buat seperti ini.



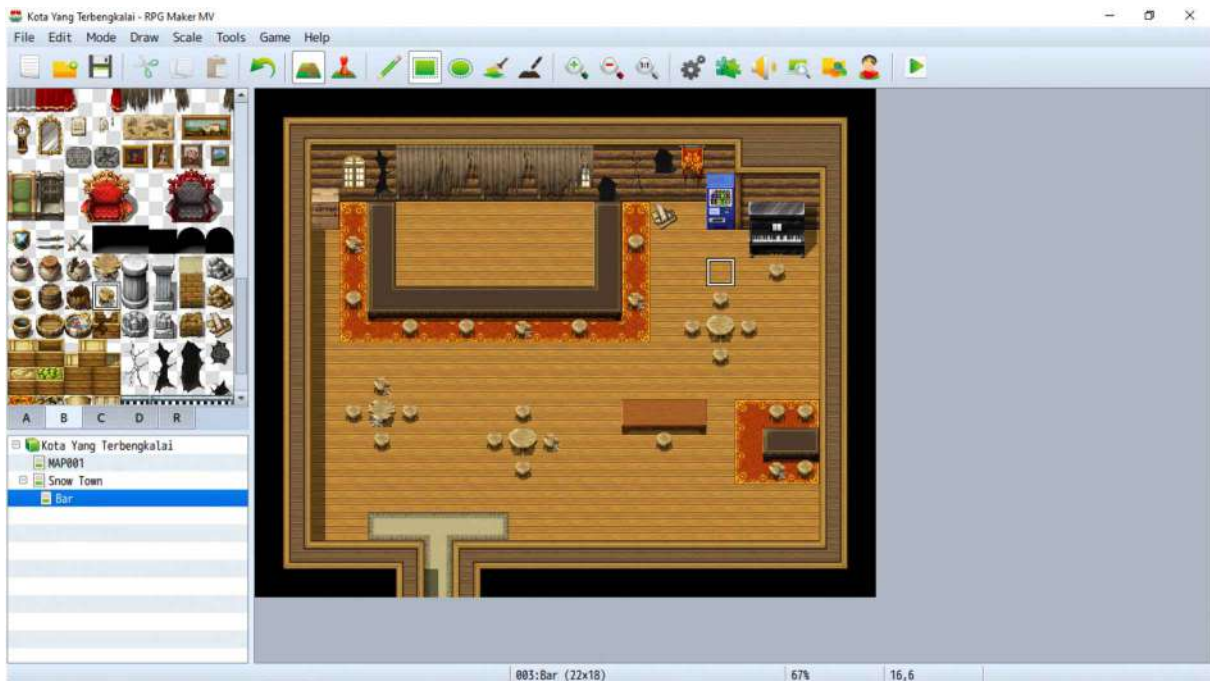
Gambar 3.52. Membuat lantai *map Bar*

Setelah itu tambahkan objek seperti gambar di bawah ini.



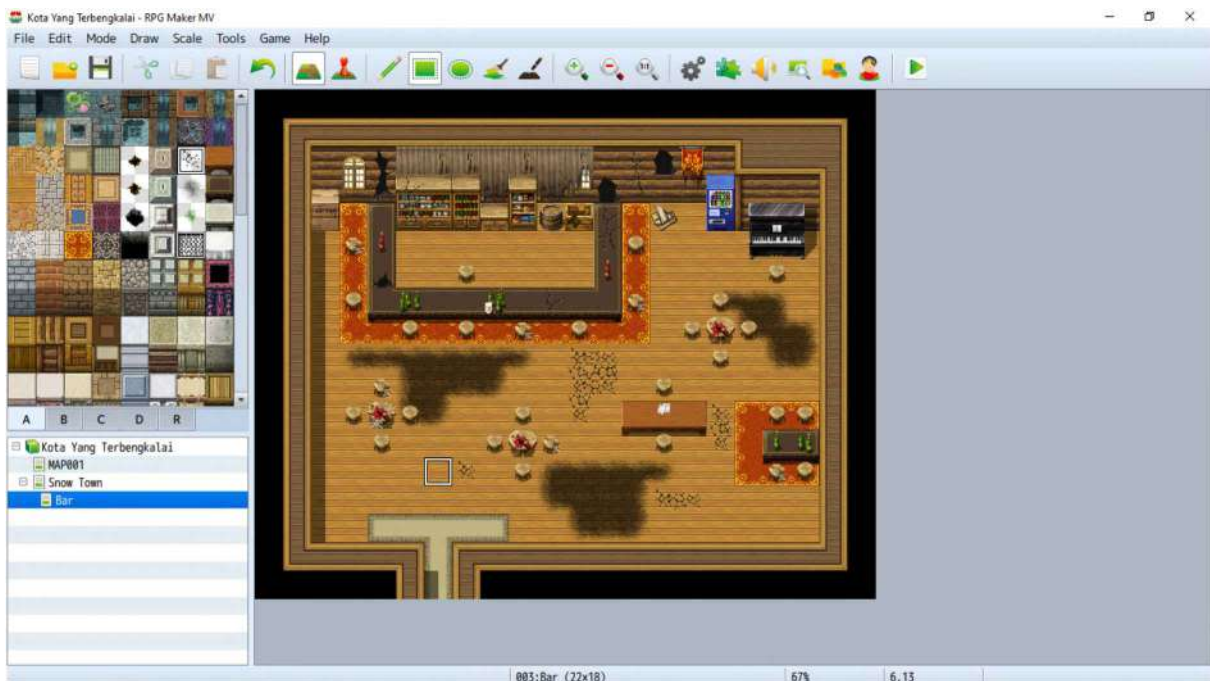
Gambar 3.53. Menambahkan objek yang menempel dinding

Kemudian tambahkan meja, kursi, dan karpet seperti ini.



Gambar 3.54. Menambahkan objek di *map* Bar

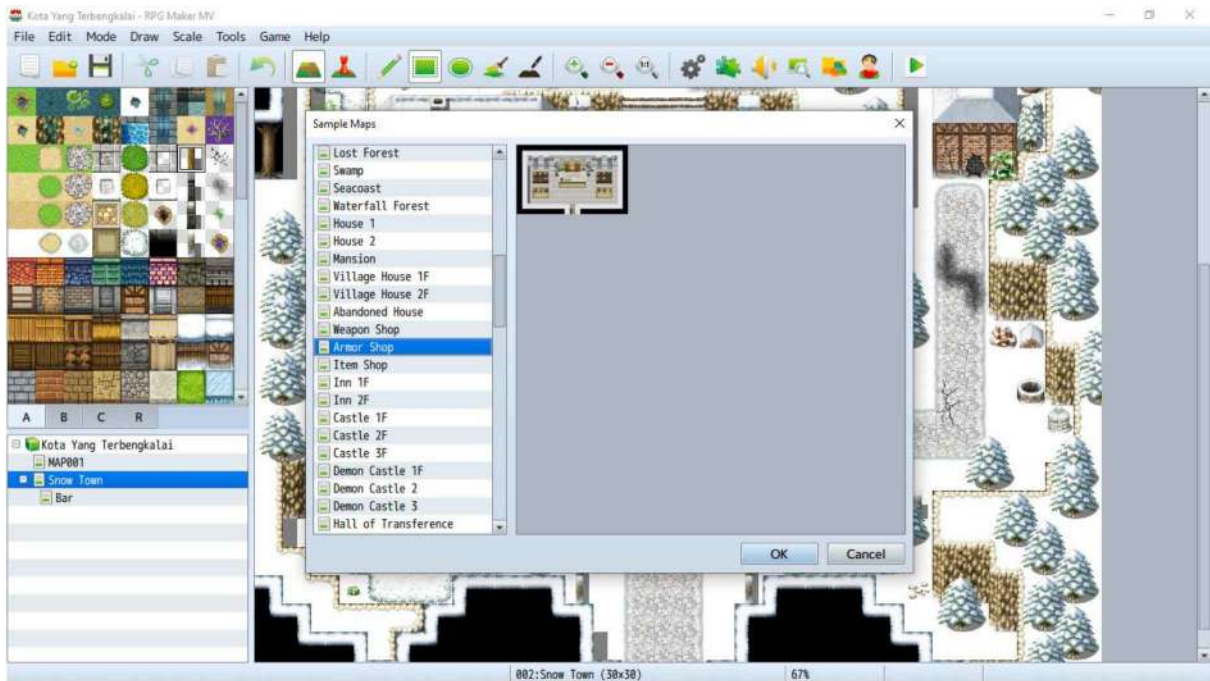
Setelah menambah meja, kursi, dan karpet, kemudian tambahkan objek pendukung seperti ini.



Gambar 3.55. Menambahkan objek pendukung

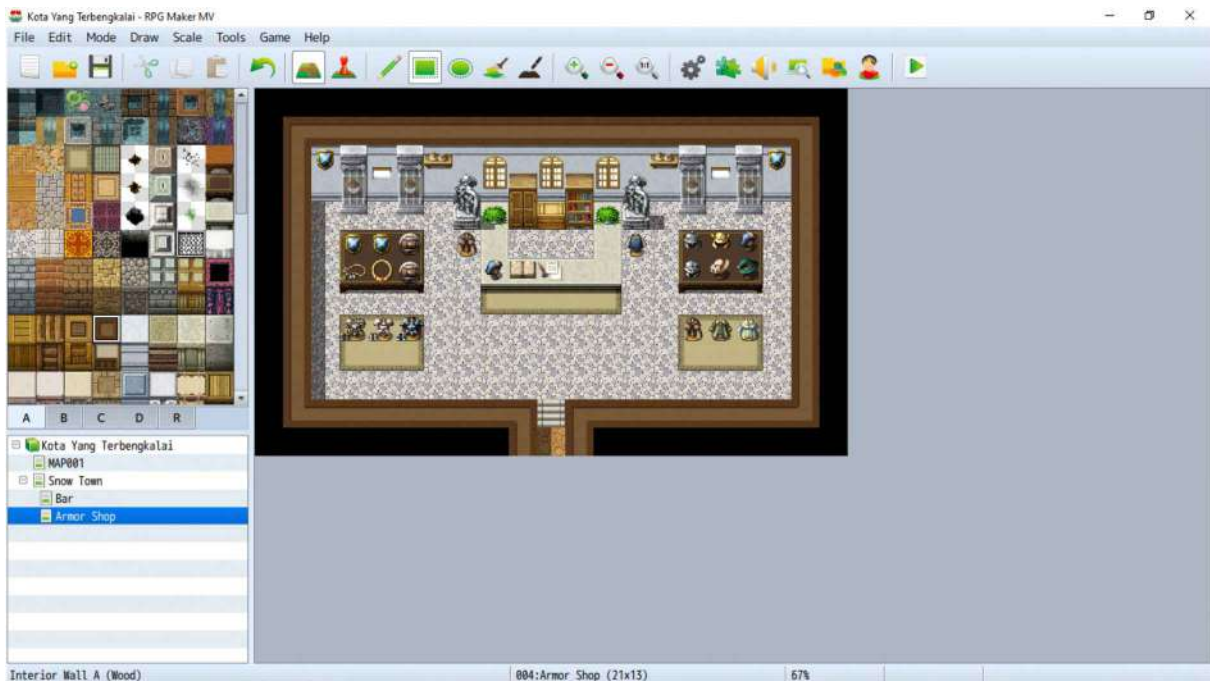
Armor Shop

Setelah *map* Bar selesai, selanjutnya buat *map* Armor Shop di bawah *map* Snow Town dengan cara klik kanan pada Snow Town lalu klik *Load* dan pilih *Armor Shop*.



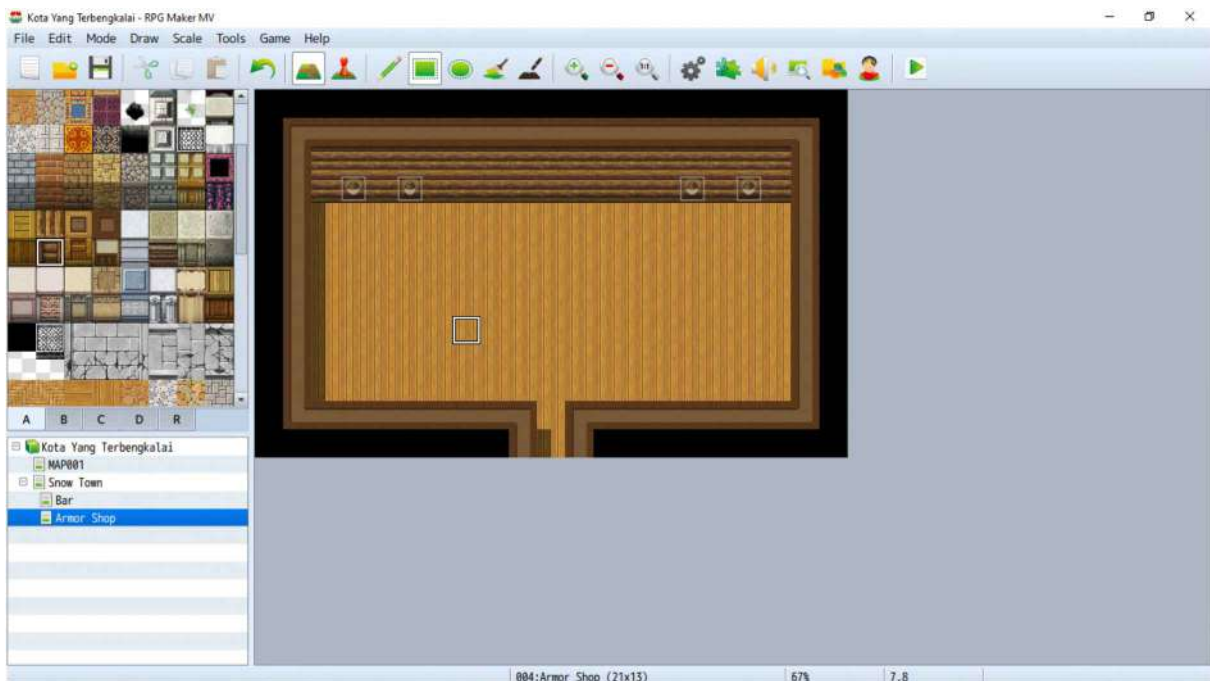
Gambar 3.56. Membuat *map* baru

Lakukan modifikasi pada *map* ini dengan mengubah *outline* *map*-nya dengan *tile Interior Wall A (Wood)* seperti ini.



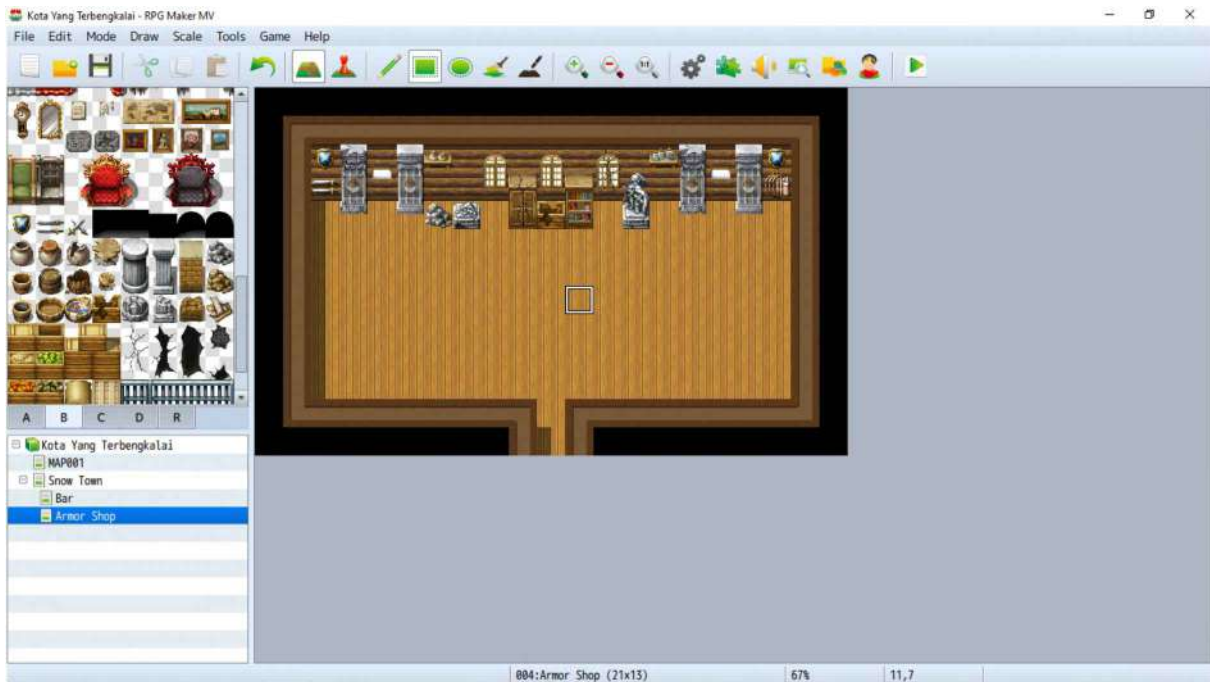
Gambar 3.57. Mengubah *outline map* *Armor Shop*

Selanjutnya hapus terlebih dahulu objeknya dan mengganti dindingnya dengan *tile Wall J (Log)* dan lantainya dengan *Wood Floor D*. Untuk objek lampu biarkan saja seperti ini.



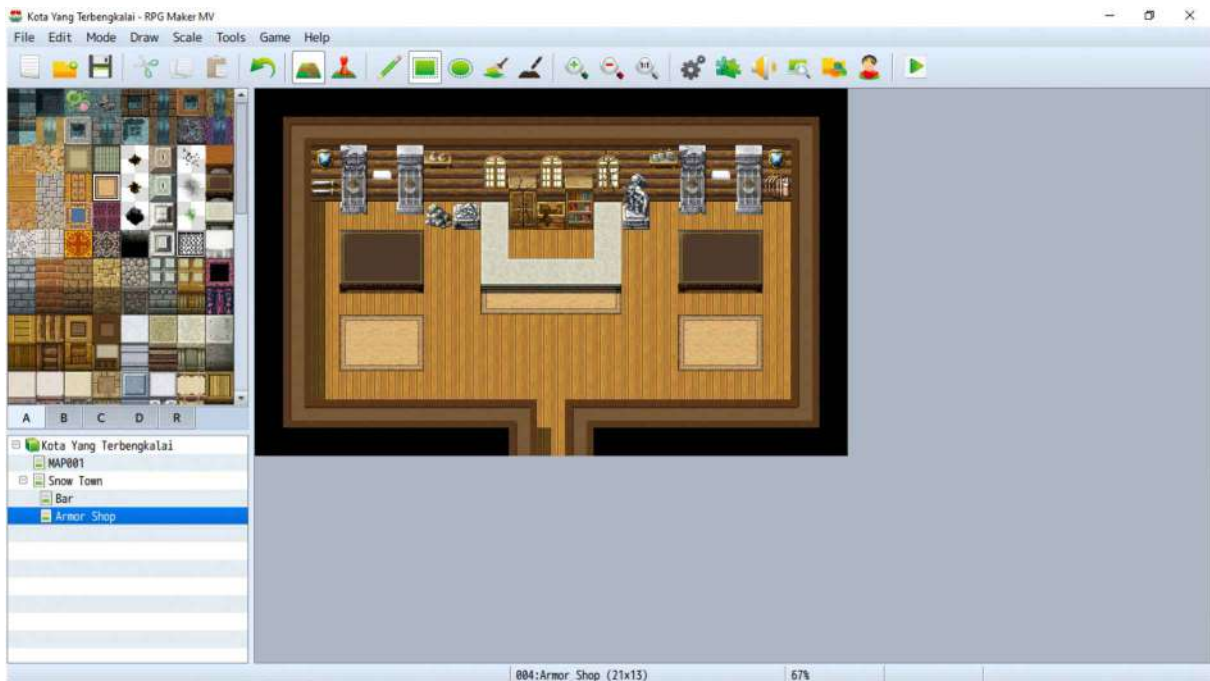
Gambar 3.58. Mengganti *tile* dinding dan lantai *Armor Shop*

Setelah mengganti dinding dan lantai, tambahkan objek yang menempel dinding.



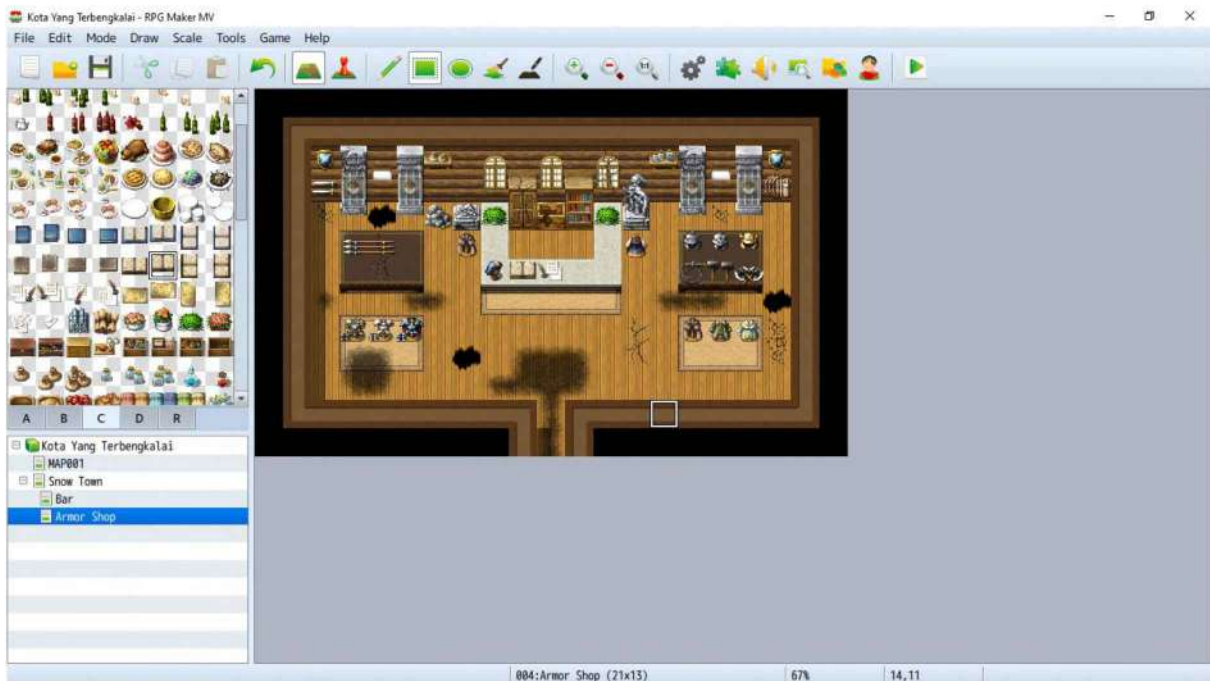
Gambar 3.59. Menambahkan objek di dinding *Armor Shop*

Selanjutnya tambahkan objek meja dan karpet.



Gambar 3.60. Menambahkan meja dan karpet di *Armor Shop*

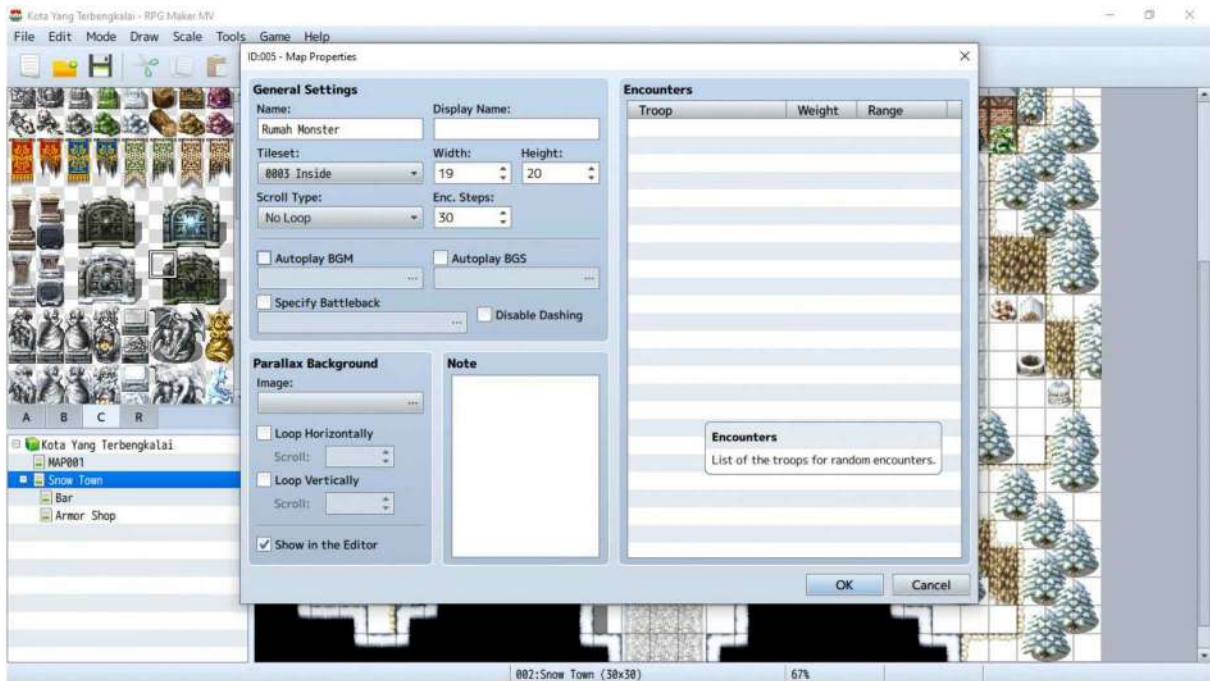
Yang terakhir adalah menambah objek dan efek pecahan. Hasilnya akan seperti ini.



Gambar 3.61. Menambahkan objek di *Armor Shop*

Rumah Monster

Yang terakhir adalah membuat *map* untuk Rumah *Monster*. Caranya sama, yaitu klik kanan di *Snow Town*, kemudian klik *New*.

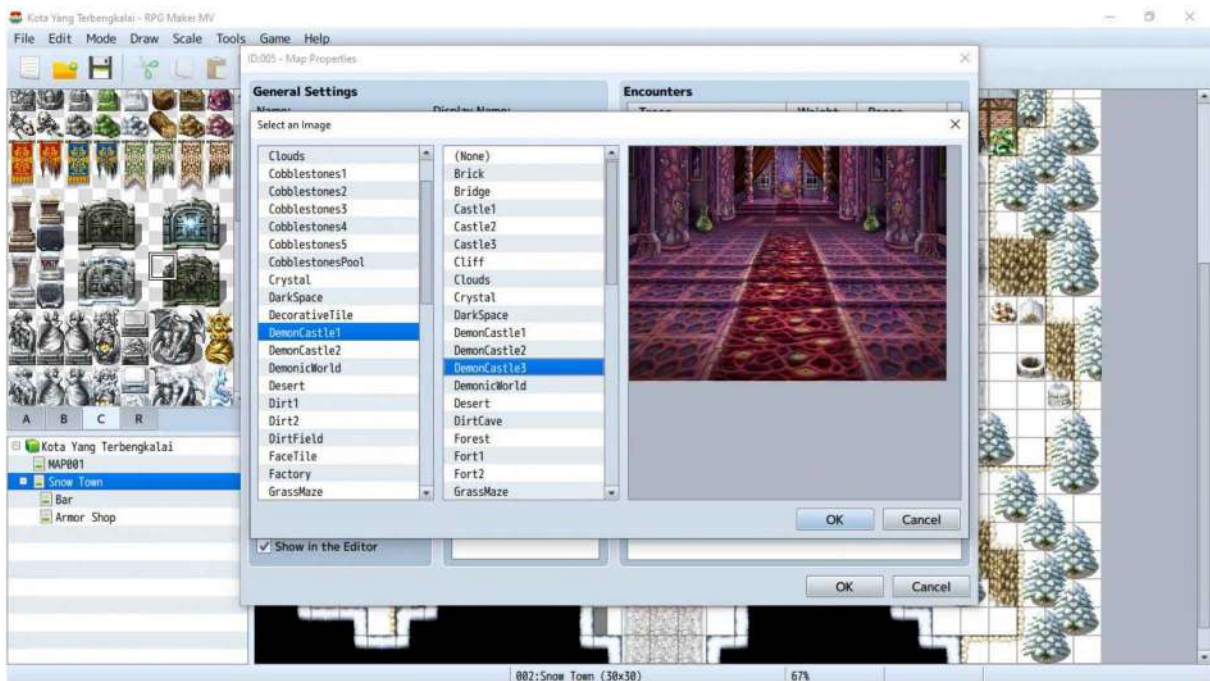


Gambar 3.62. Membuat *map* Rumah *Monster*

Setelah itu atur *Autoplay BGM* dan *Specific Battleback*-nya seperti ini.



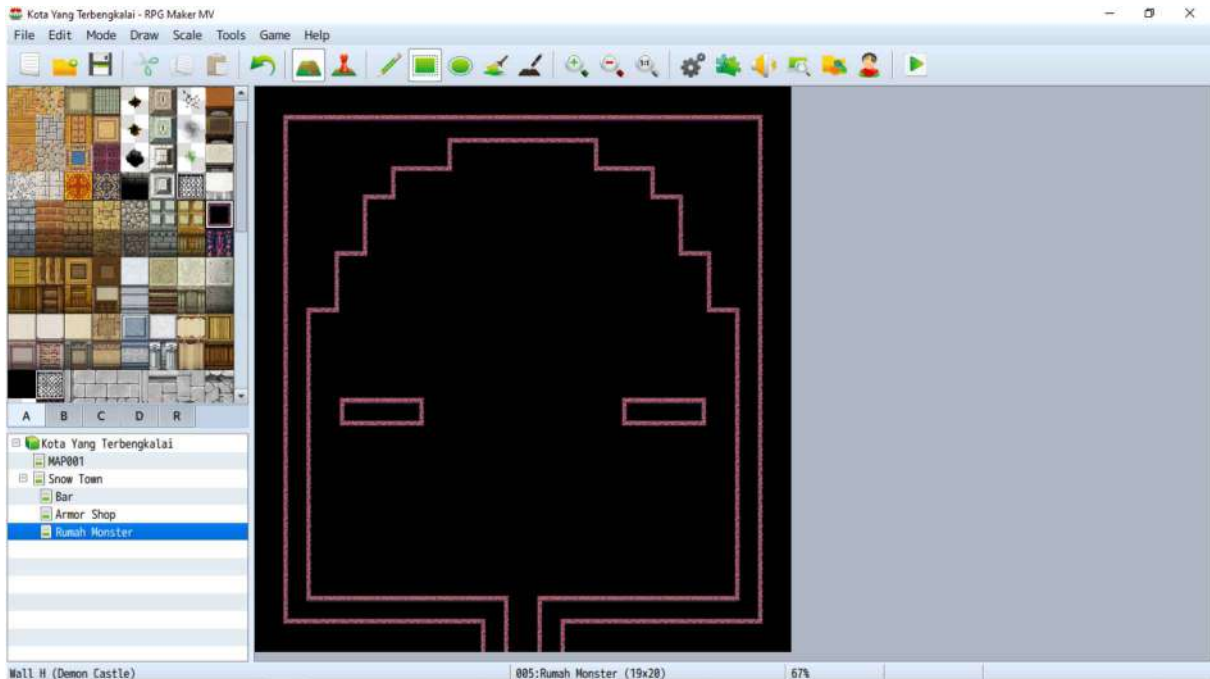
Gambar 3.63. Mengatur *Autoplay BGM*



Gambar 3.64. Mengatur *Specific Battleback*

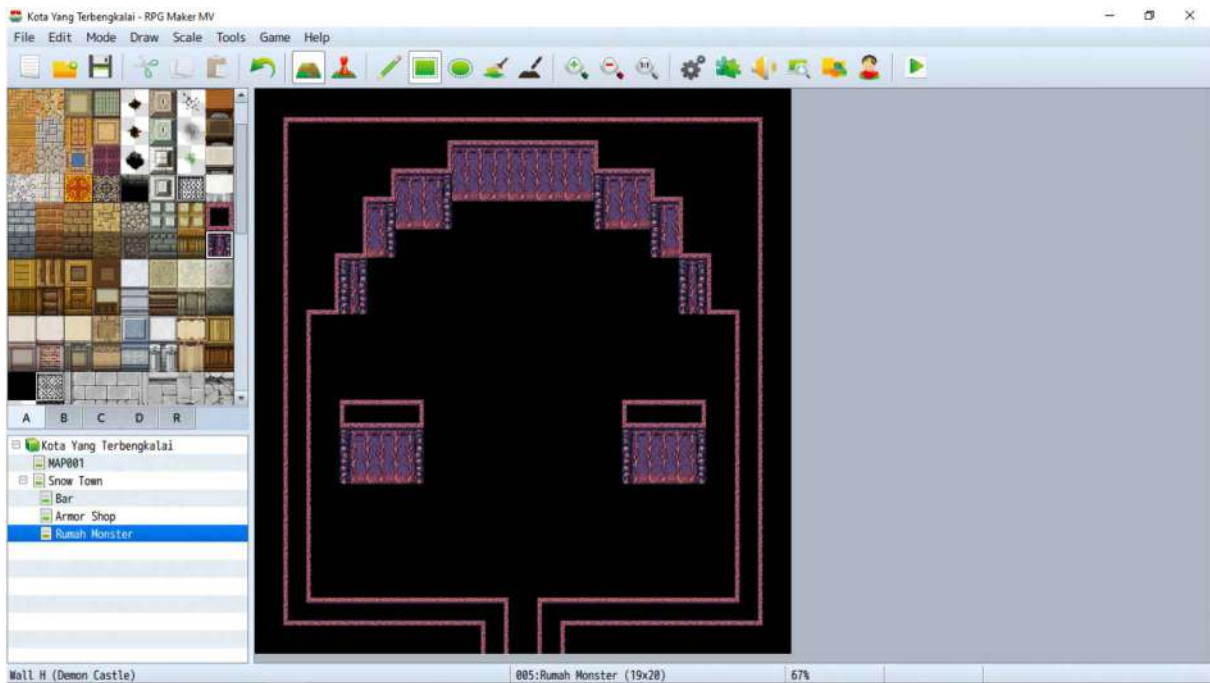
Setelah itu klik *OK*. Kemudian blok semua *map* dengan warna hitam seperti cara sebelumnya, yaitu menggunakan *tile Darkness*. Setelah selesai memblok, gunakan *tile Wall H*

(*Demon Castle*) di *tileset A* untuk membuat *outline* seperti gambar di bawah ini.



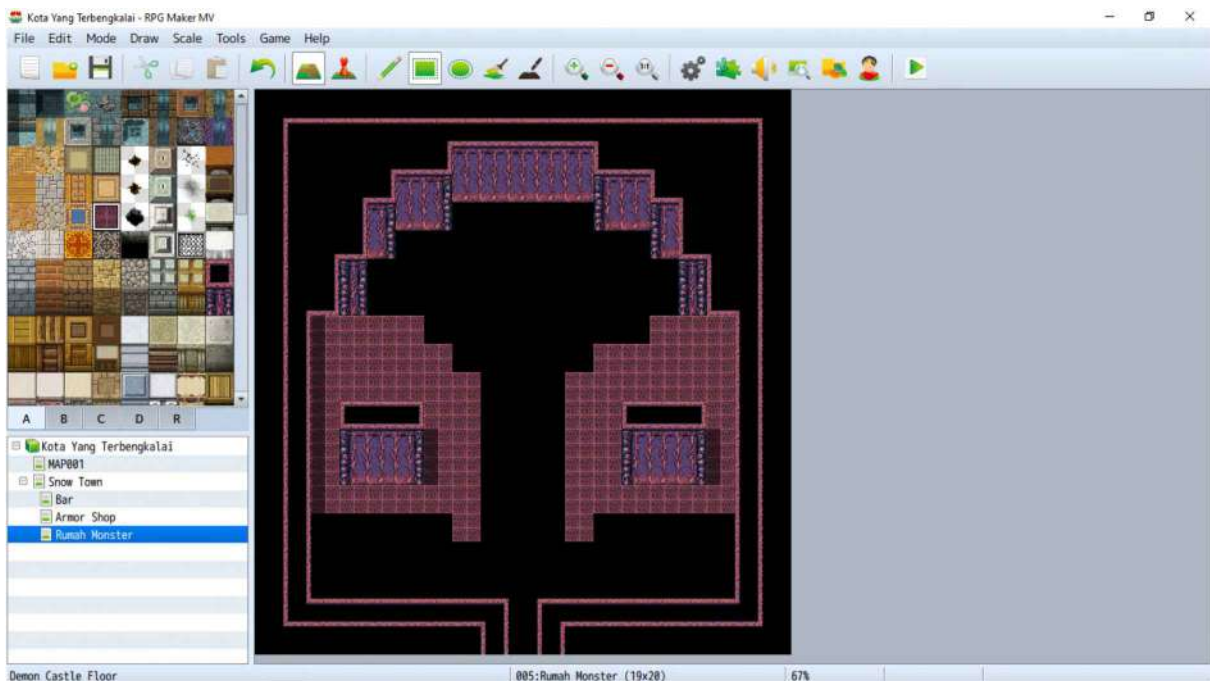
Gambar 3.65. Membuat *outline map* Rumah Monster

Kemudian buat dinding menggunakan *tile Wall H (Demon Castle)* di bawah *tile outline* seperti ini.



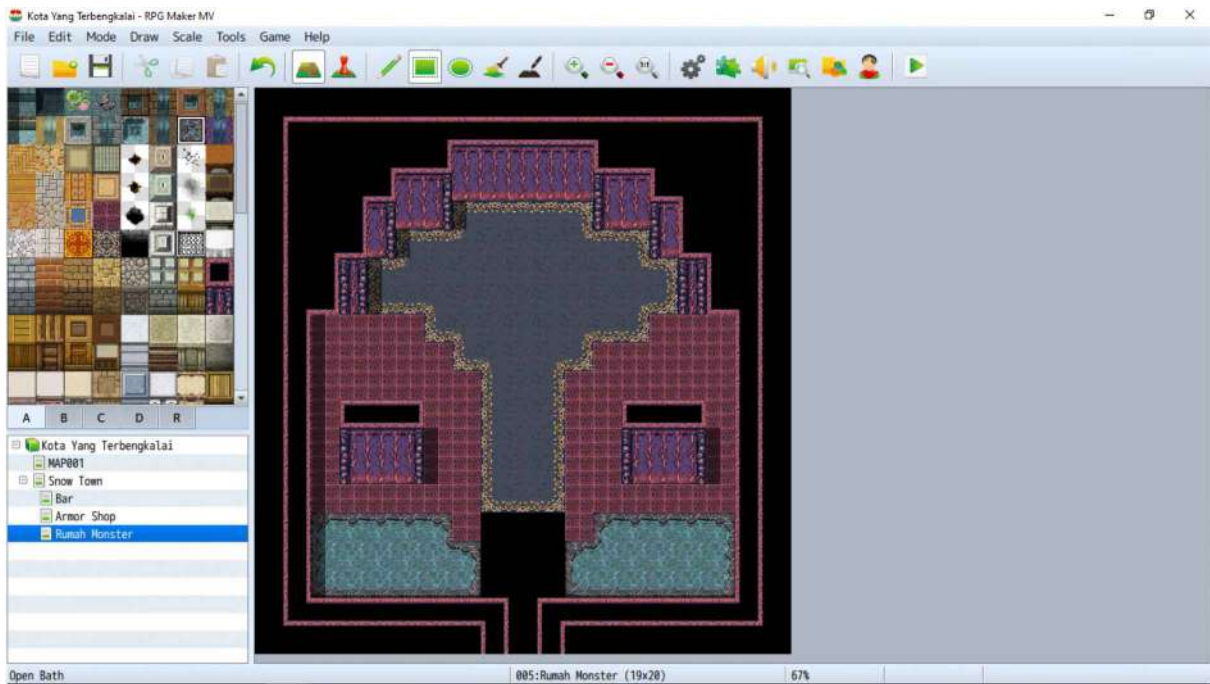
Gambar 3.66. Membuat dinding *map* Rumah Monster

Setelah itu buat lantai menggunakan *tile Demon Castle Floor* di *tilesets A* seperti ini.



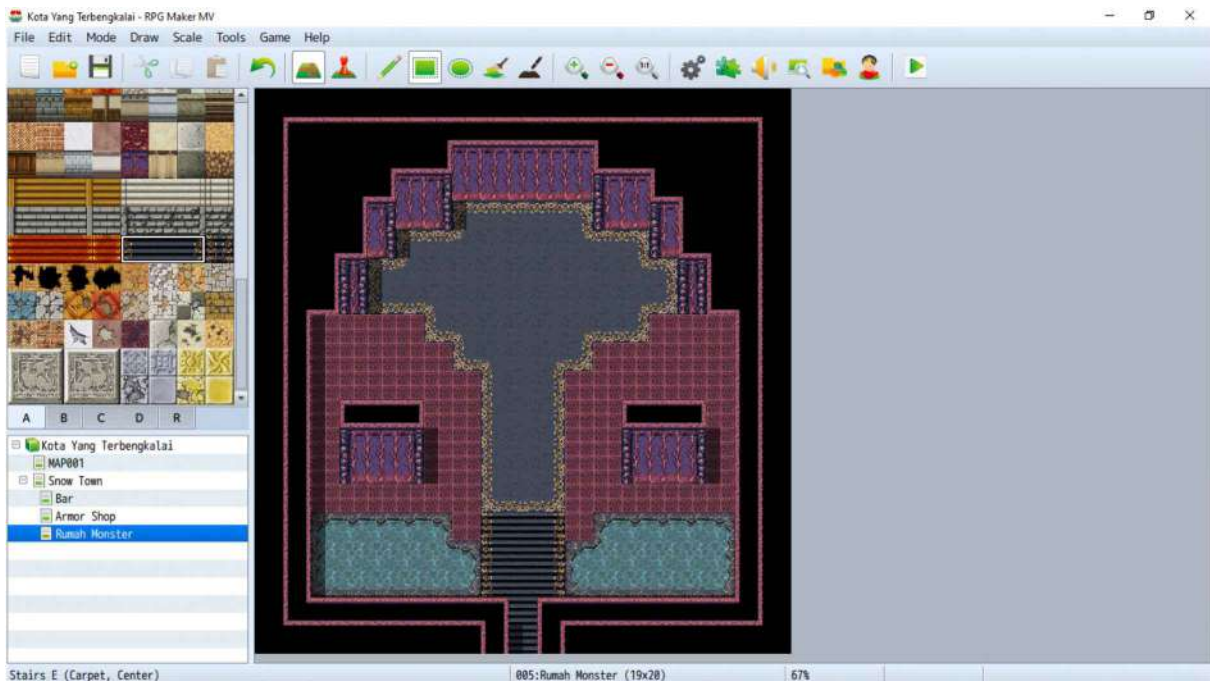
Gambar 3.67. Membuat lantai *map* Rumah Monster

Selanjutnya tambahkan karpet menggunakan *tile Rug F* dan *tile Open Bath* untuk kolam di *tilesets A* yang hasilnya akan seperti ini.



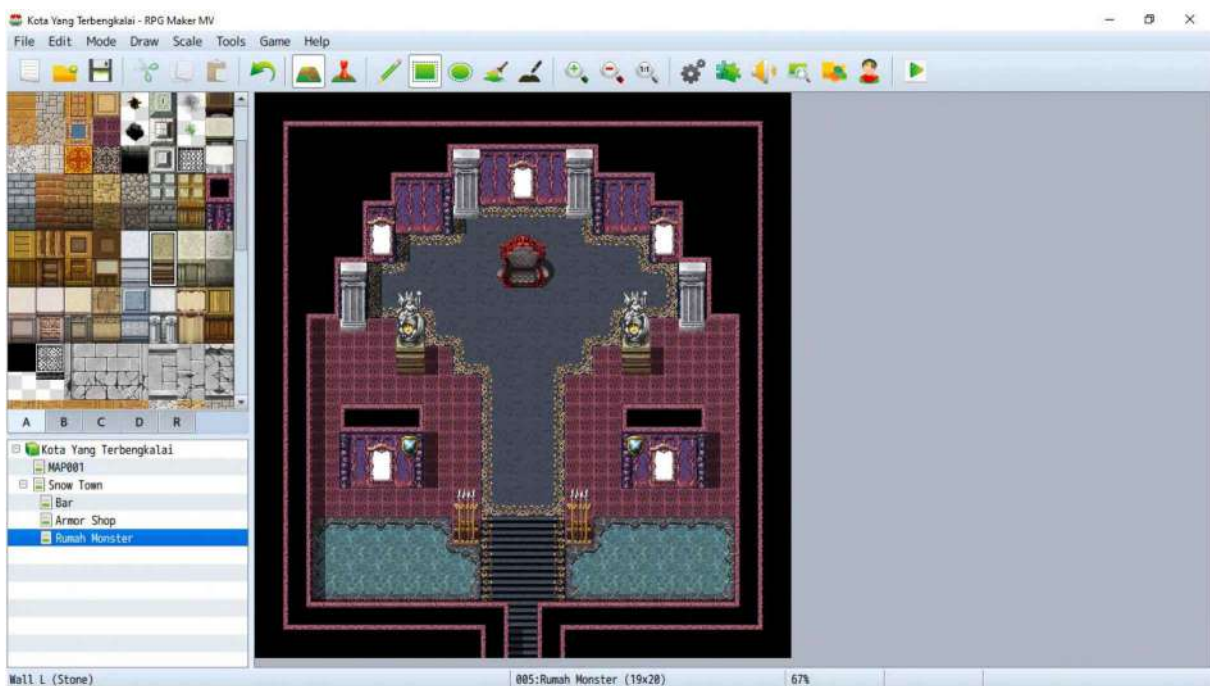
Gambar 3.68. Membuat karpet dan kolam di *map* Rumah *Monster*

Kemudian tambahkan tangga menggunakan *tile Stairs E (Carpet, Left)*, *tile Stairs E (Carpet, Center)*, dan *tile Stairs E (Carpet, Right)* di *tilesets A* yang hasilnya akan seperti ini.



Gambar 3.69. Membuat tangga di *map* Rumah *Monster*

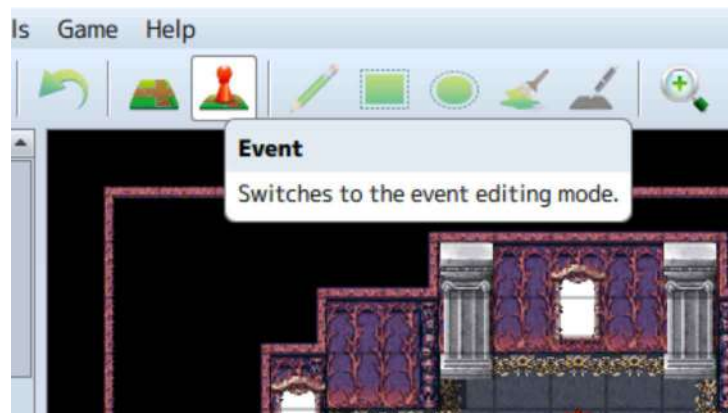
Yang terakhir adalah menambahkan objek yang hasilnya akan seperti ini.



Gambar 3.70. Menambahkan objek di *map* Rumah *Monster*

Membuat Event

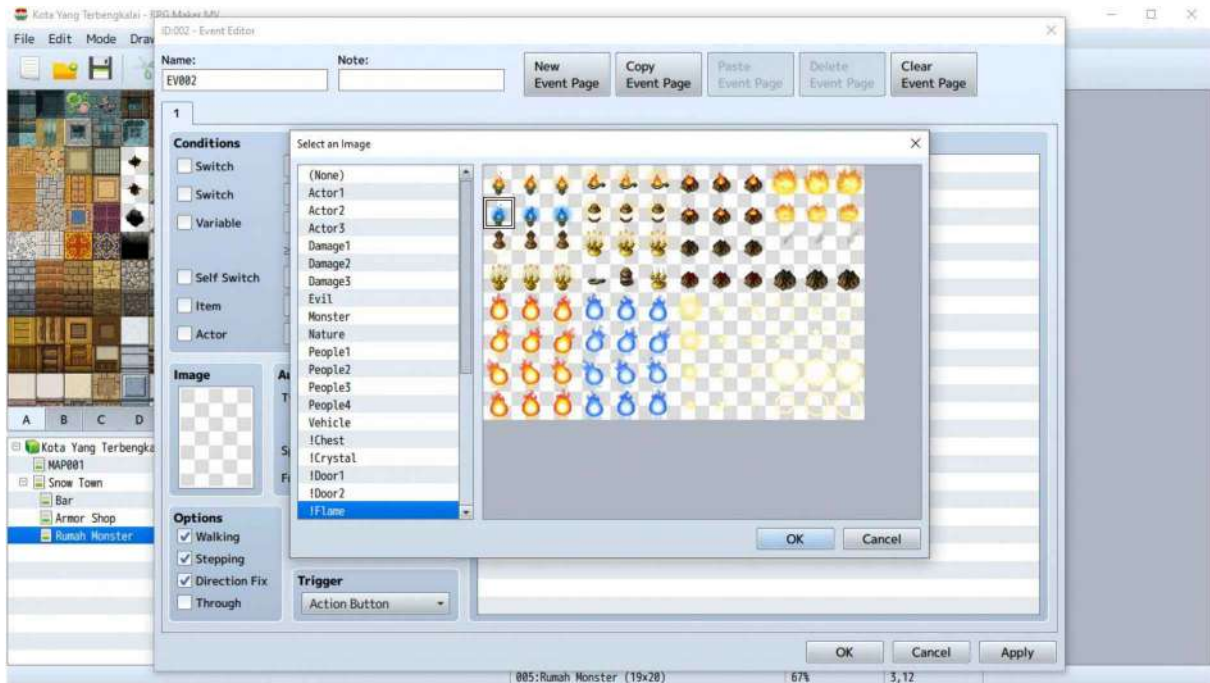
Setelah pembuatan *map* selesai, langkah selanjutnya adalah membuat sebuah *event*. *Event* bisa berupa percakapan, pintu untuk transfer antar *map*, membuat animasi lampu, dan lain-lain. Klik tombol event untuk masuk ke *mode edit event*.



Gambar 3.71. Mengubah ke *mode edit event*

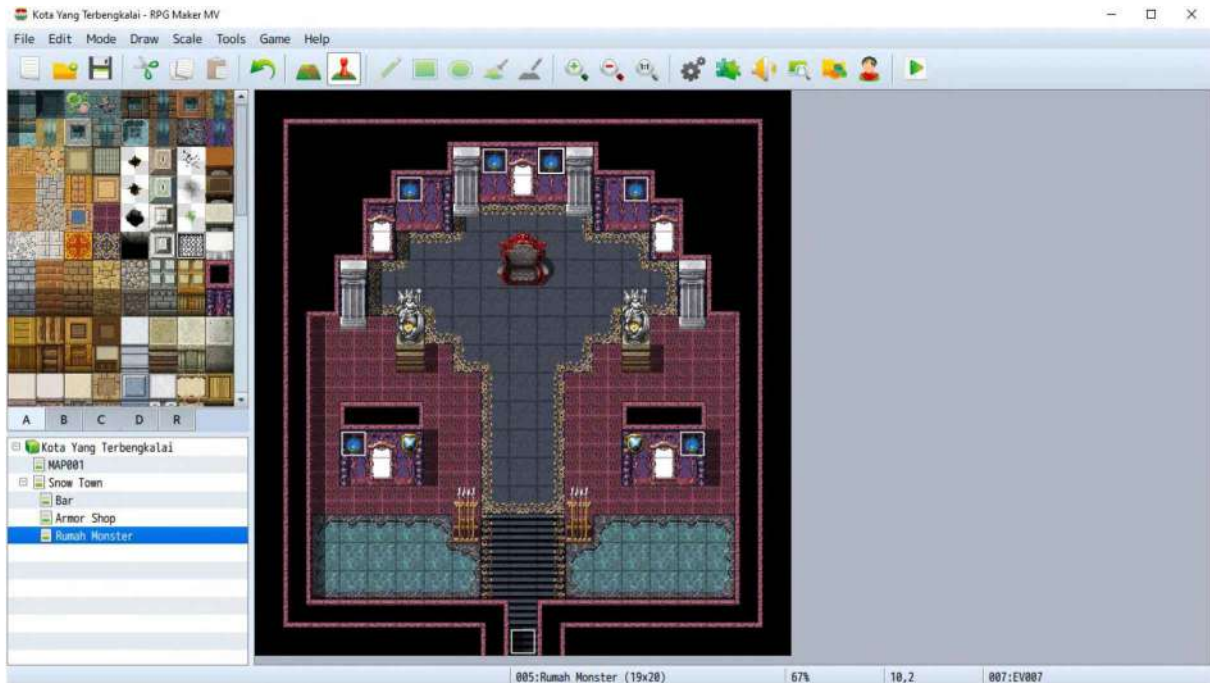
Membuat Lampu

Setelah masuk ke *mode edit event*, klik dua kali pada koordinat (3,12) lalu ganti *Priority*-nya menjadi *Same as Characters*. Kemudian centang *Stepping* dan *Direction Fix* pada kolom *Options* di kiri bawah lalu klik pada kolom *image* dan pilih *!Flame* seperti gambar di bawah ini.



Gambar 3.72. Membuat *event* lampu

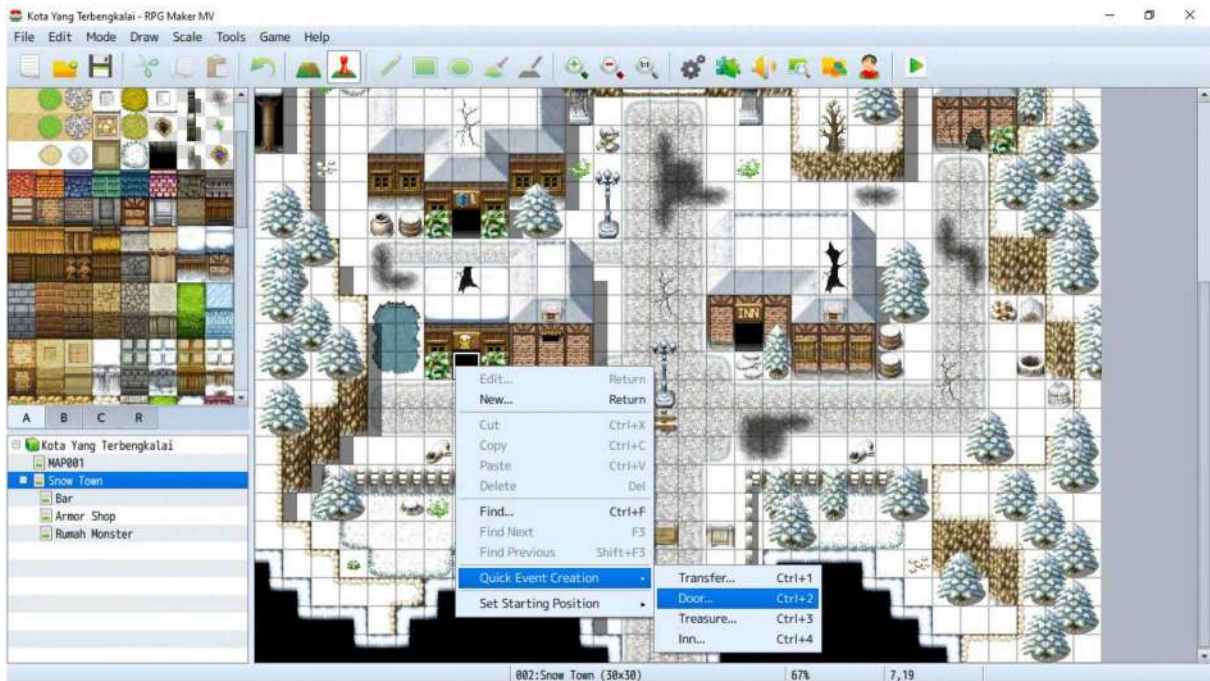
Lakukan hal yang sama untuk membuat *event* lampu di koordinat (15,12), (5,3), (13,3), (8,2), dan (10,2) sehingga hasilnya akan seperti gambar di bawah ini.



Gambar 3.73. Membuat beberapa *event* lampu

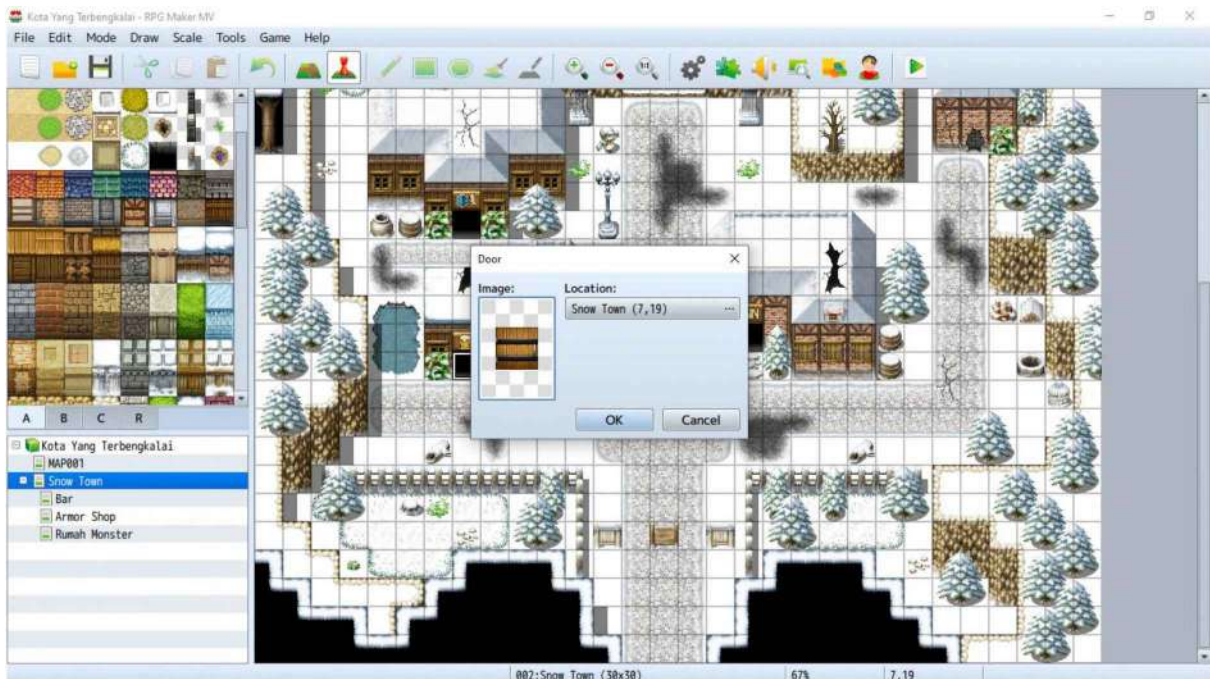
Membuat Pintu Terbuka

Klik *map Snow Town* di sebelah kiri sampai *map* berganti tampilan menjadi *map Snow Town*. Kemudian klik kanan pada koordinat (7,19) lalu pilih *Quick Event Creation* → *Door* atau dengan tombol Ctrl+2.



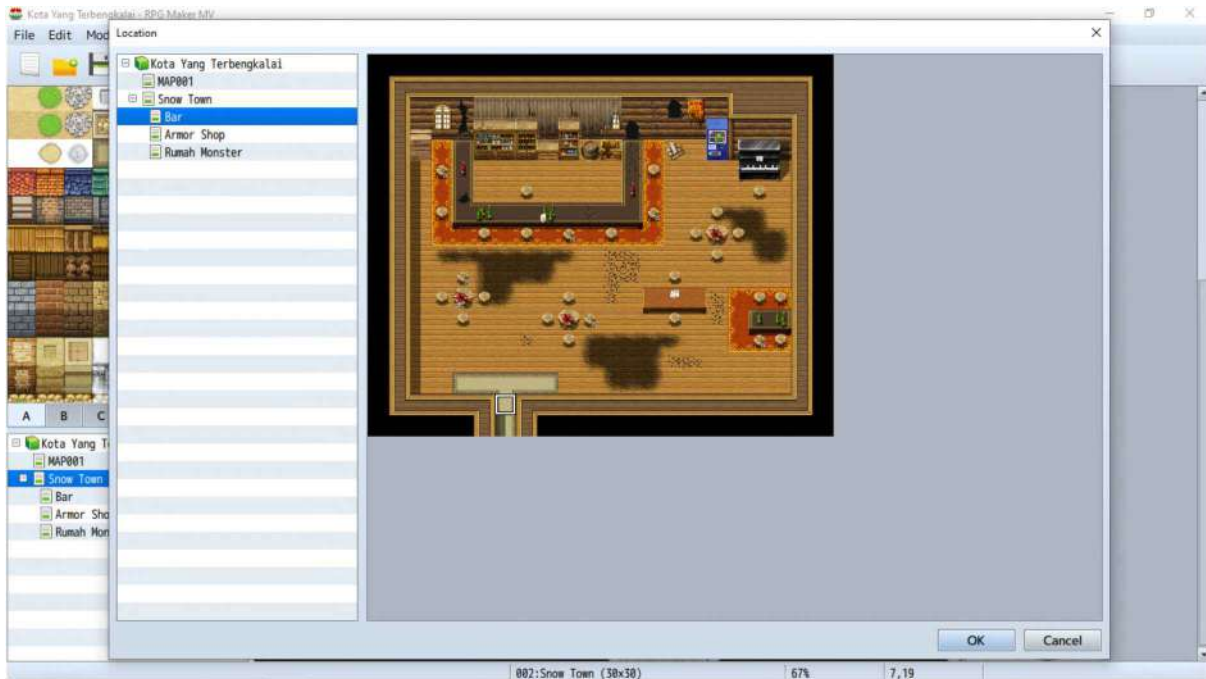
Gambar 3.74. Membuat *event* pintu

Kemudian akan muncul tampilan seperti ini.



Gambar 3.75. Tampilan *event* pintu

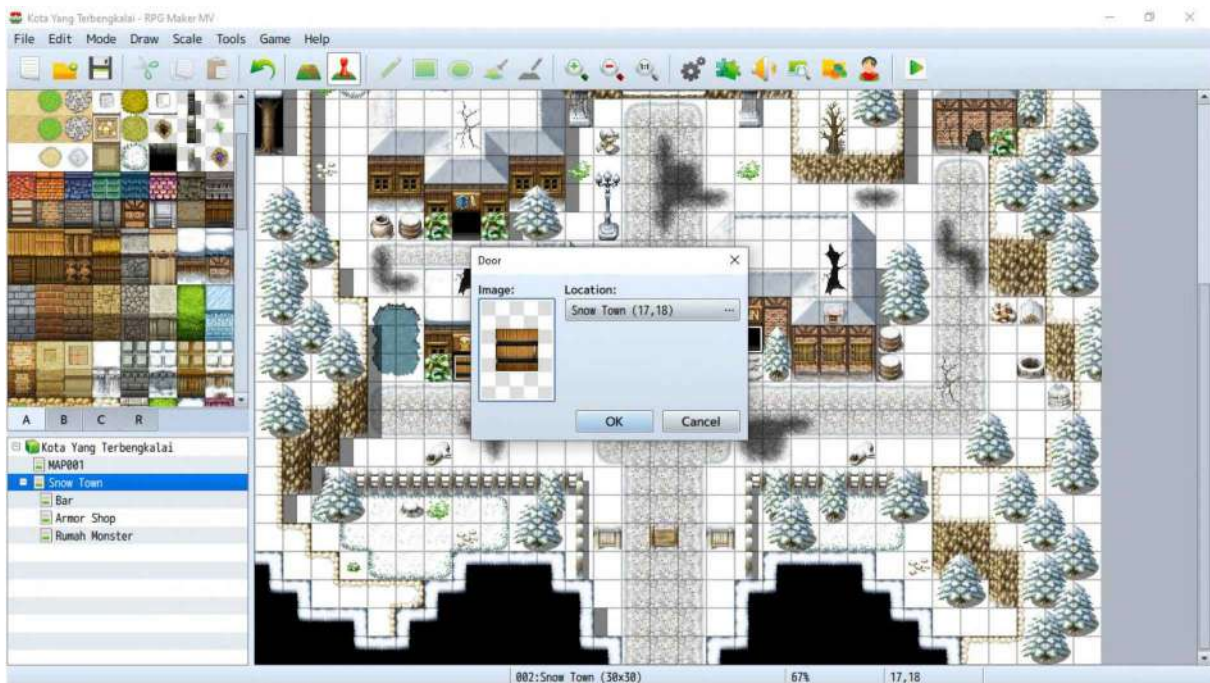
Klik tulisan *Snow Town* (7,19) dan ubah koordinatnya menjadi seperti ini kemudian itu klik *OK*.



Gambar 3.76. Membuat *event* pintu ke Bar

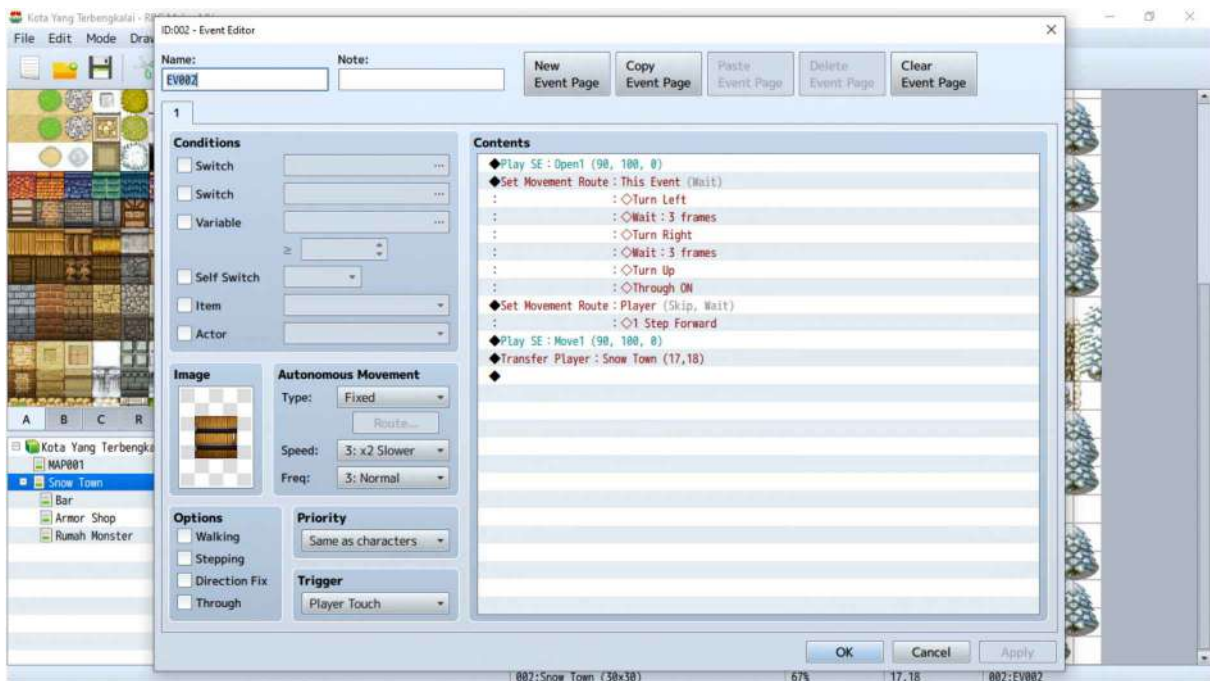
Membuat Pintu Terkunci

Selanjutnya buat pintu pada bangunan Penginapan yang dalam kondisi terkunci. Pemain tidak akan berpindah ke *map* lain jika mendekati pintu Penginapan dan akan muncul tulisan “Terkunci.” Caranya sama seperti membuat *event* pintu sebelumnya, yaitu *Quick Event Creation* → *Door* atau dengan tombol *Ctrl+2* di koordinat (17,18). Klik *OK* saja jika muncul tampilan seperti di bawah ini.



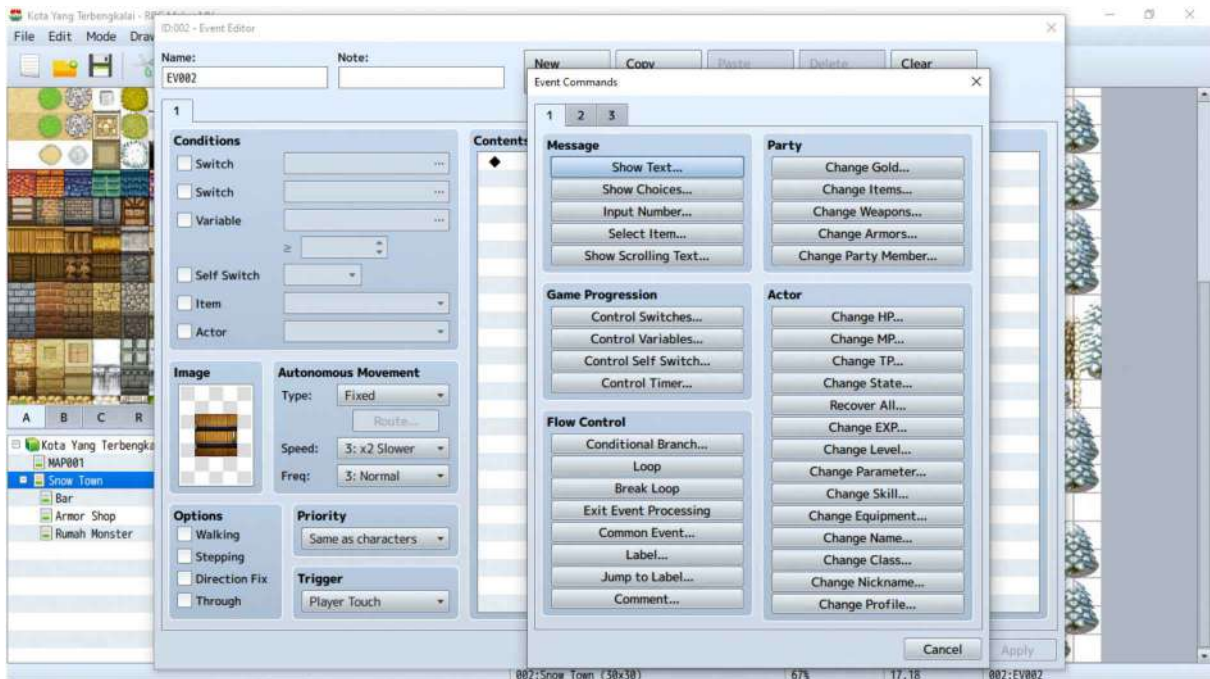
Gambar 3.77. Mengunci pintu bangunan Penginapan

Kemudian klik dua kali pada pintu di koordinat (17,18) yang akan memunculkan jendela seperti ini.



Gambar 3.78. Mengubah *event* pintu Penginapan

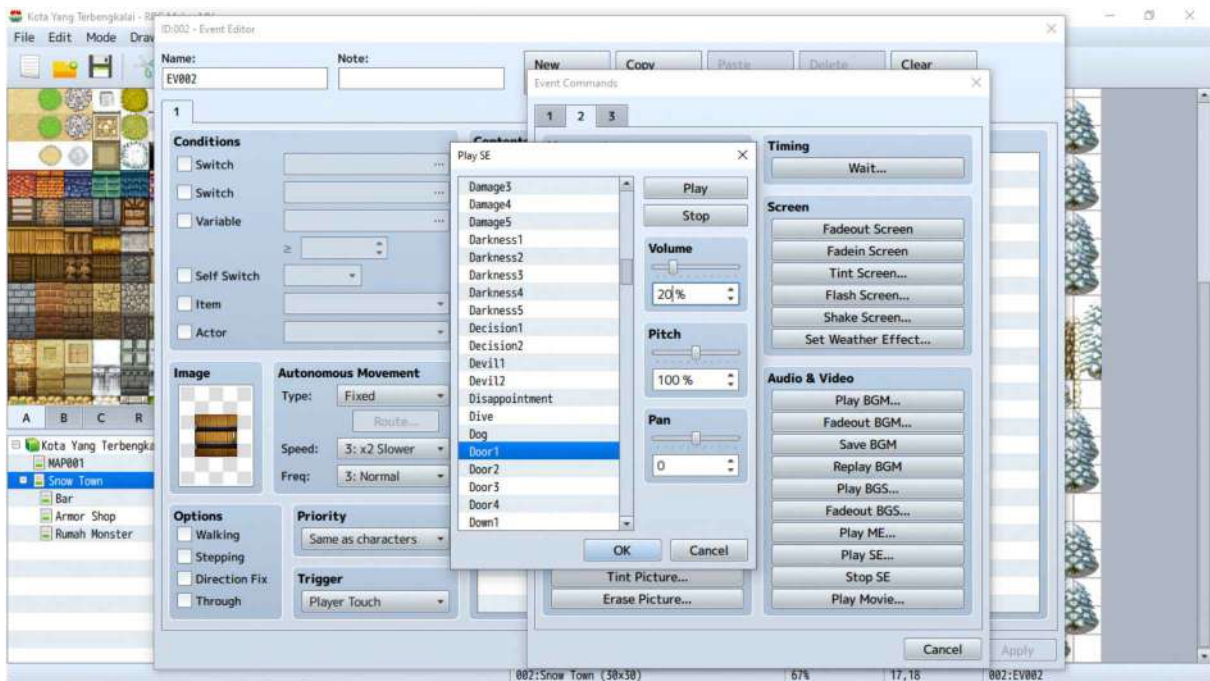
Kemudian hapus semua yang ada di kotak *Contents*. Setelah itu klik baris pertama maka akan muncul jendela seperti ini.



Gambar 3.79. Tampilan menu *Event Editor*

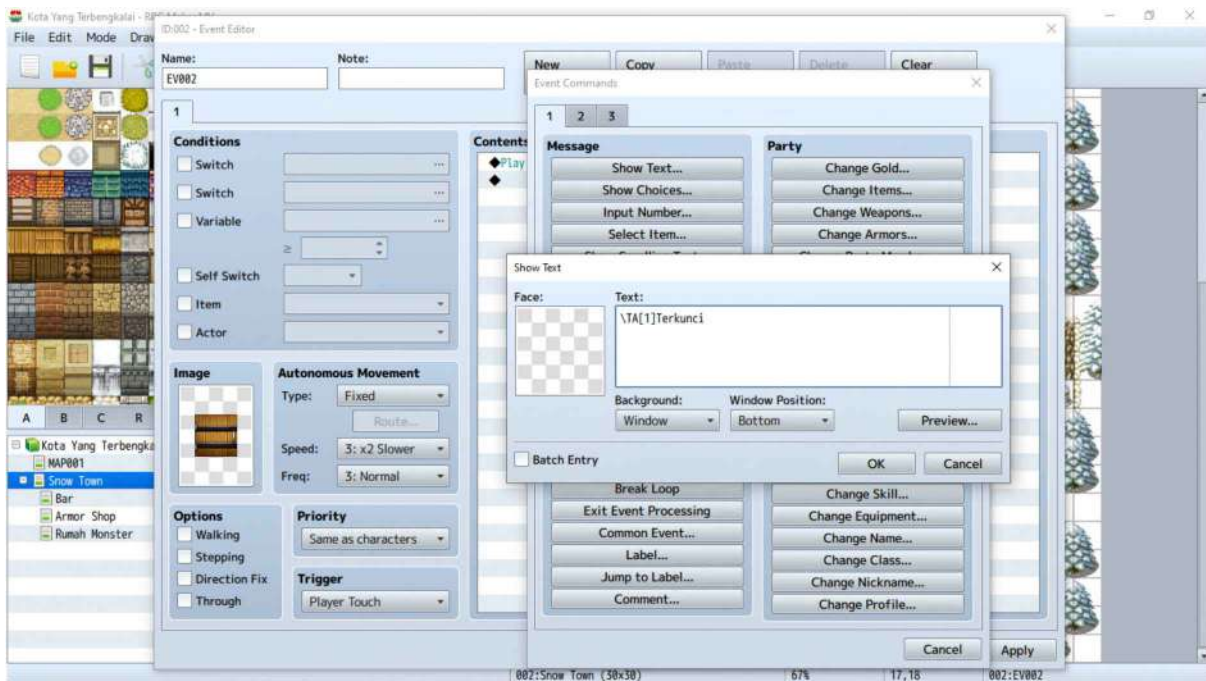
Ini adalah *Event Editor*. Jendela ini berfungsi untuk membuat *event* dalam *game*, bisa dikatakan menu ini adalah menu utama dalam pembuatan *game* menggunakan RPG Maker MV.

Klik pada halaman dua *Event Editor* dan klik tulisan *Play SE* di kolom *Audio & Video*. Kemudian pilih *Door1* dan atur *volume*-nya menjadi 20%, setelah itu klik *OK*.



Gambar 3.80. Tampilan menu *Play SE* di *Event Editor*

Selanjutnya klik halaman satu *Event Editor* dan klik tulisan *Show Text* di kolom *Message*. Kemudian tulis seperti gambar di bawah ini. Setelah itu klik OK.



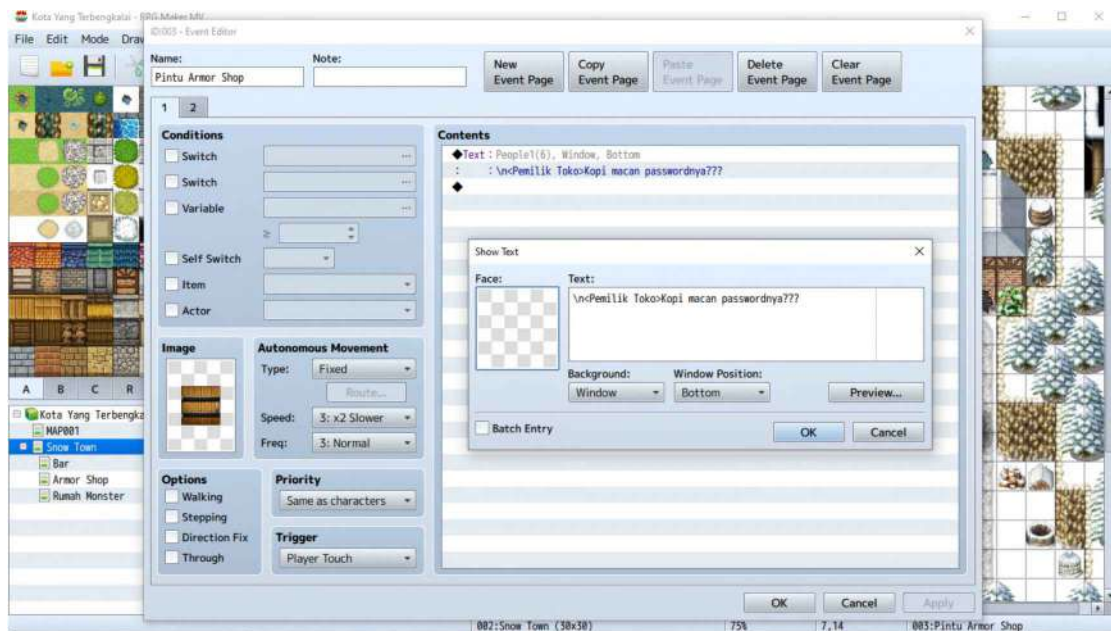
Gambar 3.81. Tampilan menu *Show Text* di *Event Editor*

Semua kode di RPG Maker MV diawali dengan simbol \. Tulisan TA dalam \TA[1] memiliki arti *text align*. Sedangkan angka di sebelahnya merupakan kode 1 yang berarti rata tengah, 0 berarti rata kiri, dan 2 adalah rata kanan. Kode ini merupakan kode tambahan dari *plugin YEP_MessageCore* yang telah diinstall di awal.

Membuat Pintu Bersyarat

Selanjutnya buat pintu lagi untuk *map Armor Shop* di koordinat (7,14). Kemudian lakukan langkah yang sama seperti membuat pintu di bangunan Penginapan dan beri nama event ini dengan **Pintu Armor Shop**. Setelah menghapus semua yang ada di kotak *Condition*, klik *Copy Event Page*

kemudian *Paste Event Page* sehingga terdapat dua halaman *Event Editor*. Klik kembali halaman satu *Event Editor* kemudian *Show Text* lalu tulis seperti ini.



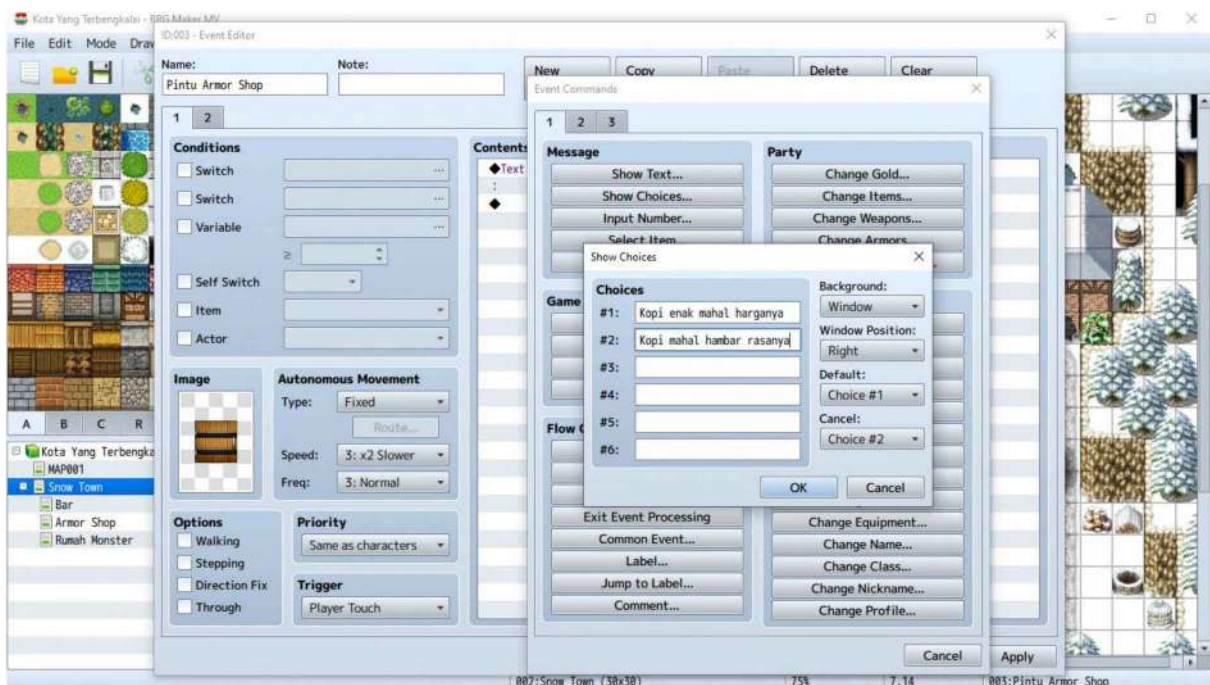
Gambar 3.82. Mengatur *event* pintu bersyarat

Klik kotak *Face*, lalu pilih *People1* nomor 6 kemudian klik *OK*.



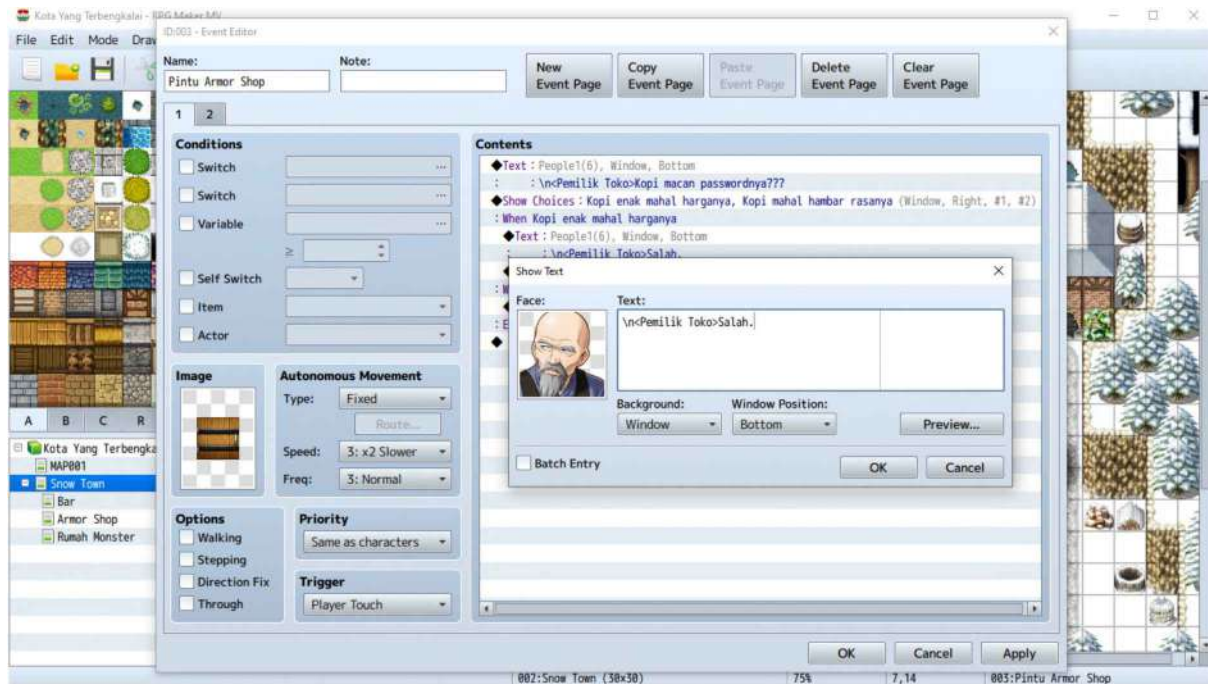
Gambar 3.83. Menambahkan wajah di menu *Show Text*

Setelah itu klik *Show Choice* dan tulis seperti ini kemudian klik *OK*.



Gambar 3.84. Menambahkan event *Show Choice*

Setelah itu klik *Show Text* di bawah baris tulisan **When Kopi enak mahal harganya** dan **When Kopi mahal hambar rasanya** dengan tulis seperti ini.

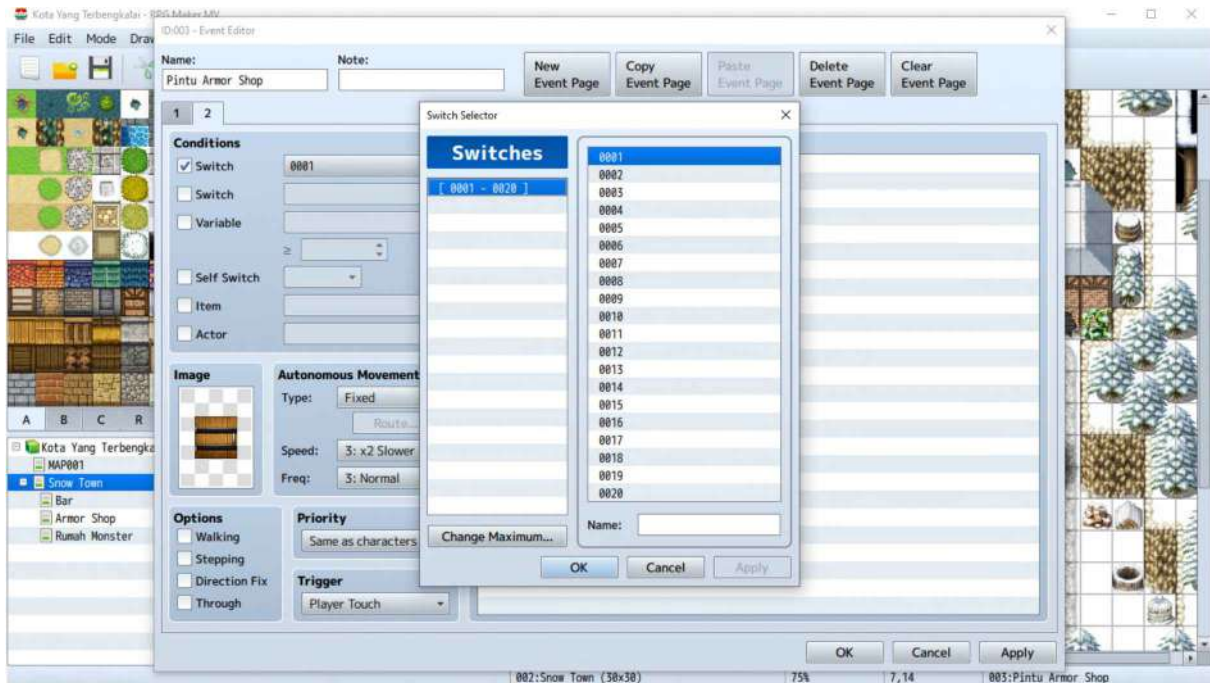


Gambar 3.85. Menambahkan hasil pilihan dari *event Show Choice*

Kode `\n<Pemilik Toko>` adalah kode yang akan memunculkan tulisan di antara `<` dan `>` di kotak yang terpisah dari teks setelahnya yang tertulis “Salah.” Kode ini digunakan untuk menulis nama karakter saat menampilkan teks.

Membuat Switch

Langkah selanjutnya adalah membuat *Switch*. Klik halaman dua kemudian centang *Switch* di kolom *Condition* di sebelah kiri lalu klik kotak 0001 yang akan muncul jendela seperti ini.



Gambar 3.86. Membuat *Switch*

Setelah muncul tampilan seperti di atas kemudian tulis *Switch*-nya seperti ini.

0001 Mencari informasi

0002 Tutorial *BlackJack*

0003 Menang *BlackJack Main Mission*

0004 Mengambil perlengkapan

0005 Dukun datang

0006 Berangkat ke rumah *monster*

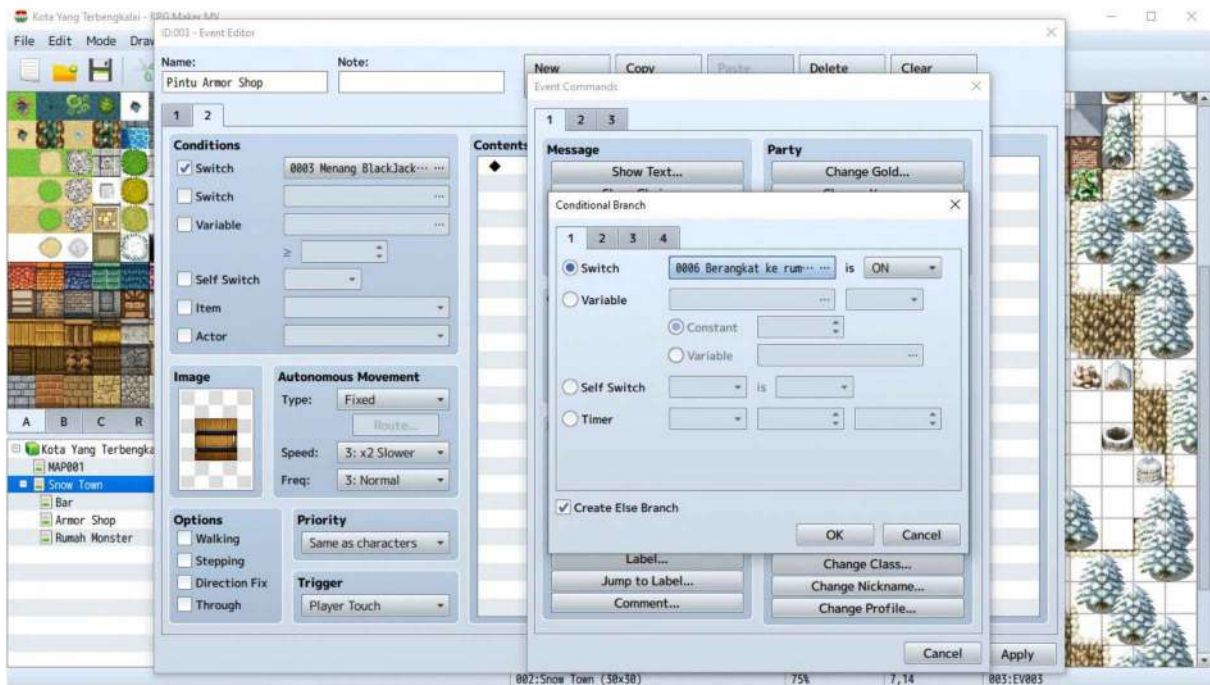
0007 *Monster* kalah



Gambar 3.87. Daftar Switch

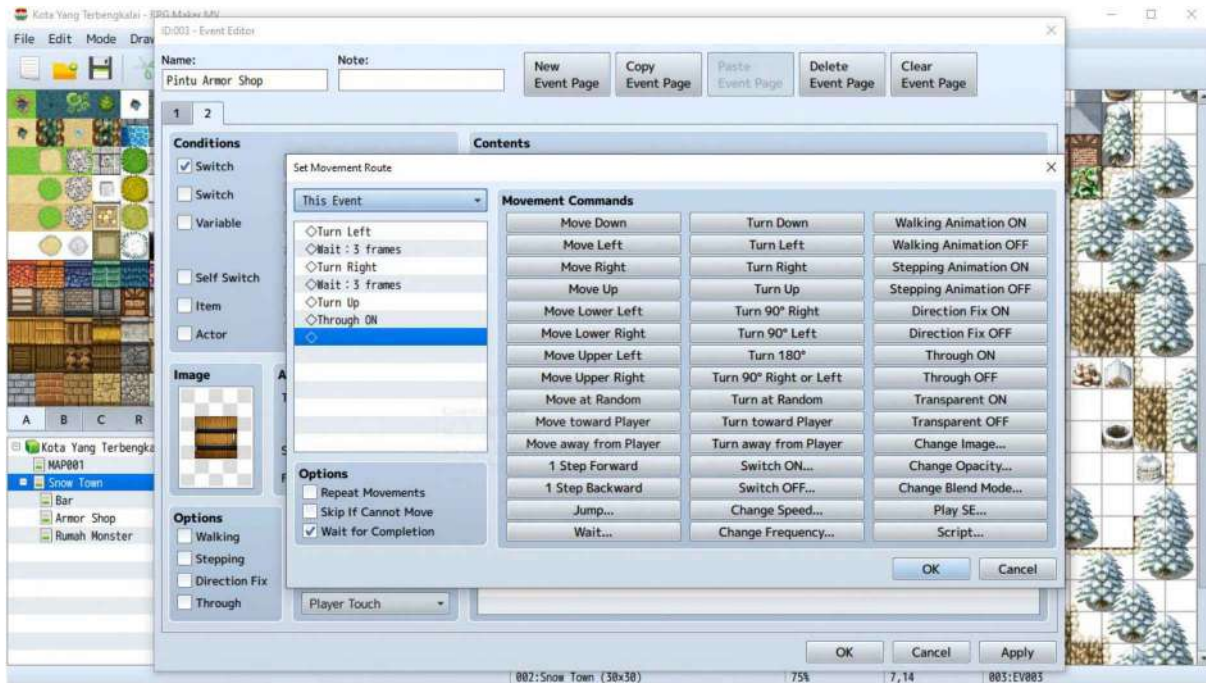
Switch tersebut akan digunakan untuk mengaktifkan *event* tertentu di *game* ini. Sebagai contoh halaman dua tidak akan aktif jika Switch Menang *BlackJack Main Mission* tidak aktif. Selanjutnya klik Switch 0003 Menang *BlackJack Main Mission* kemudian klik *Apply* lalu *OK*.

Setelah itu klik pada baris di kolom *Contents* lalu pilih *Conditional Branch*. Selanjutnya pilih Switch 0006 Berangkat ke rumah *monster* dan centang *Create Else Branch* lalu klik *OK*.



Gambar 3.88. Tampilan *Conditional Branch* pintu Armor Shop

Selanjutnya klik buat *event Set Movement Route* dan atur seperti gambar di bawah ini.



Gambar 3.89. Tampilan *event Set Movement Route* pintu *Armor Shop*

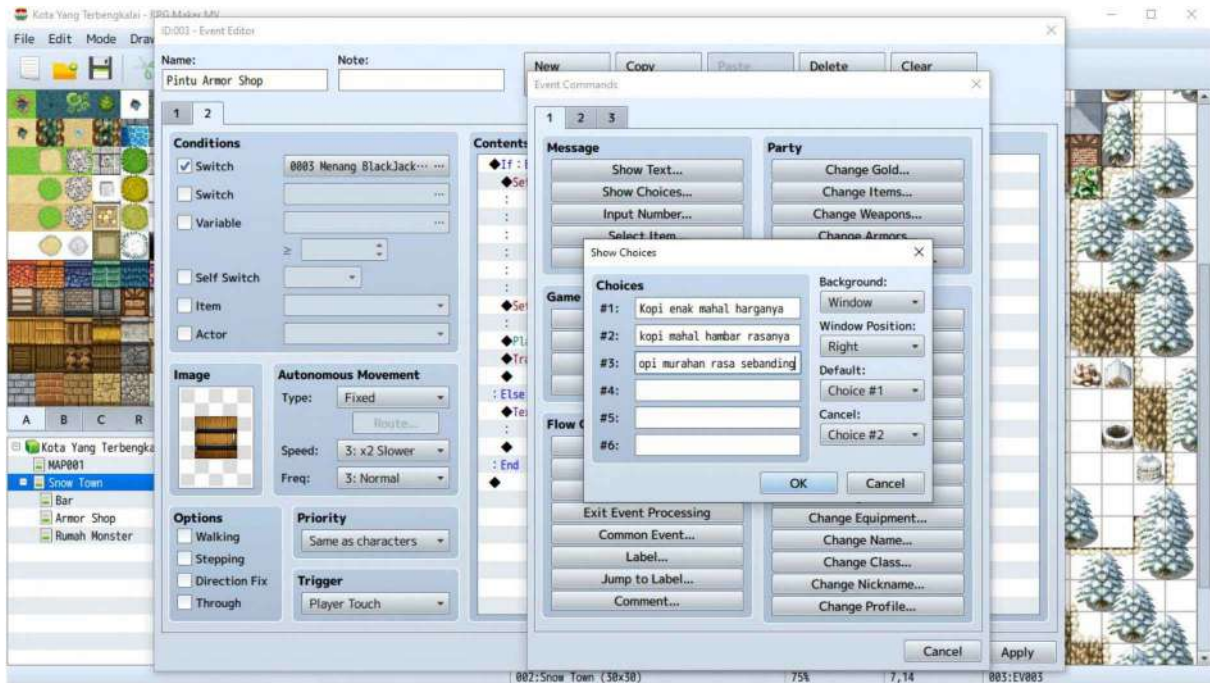
Movement Route adalah *event* untuk menggerakkan suatu *event*, apakah *event* sendiri atau bukan. Kemudian tambahkan *event* dibawah *Set Movement Route* menjadi seperti ini.

- ◆Set Movement Route : Player (**Skip, Wait**)
 - ◊ 1 Step Forward
- ◆Play SE : Move1 (40, 100, 0)
- ◆Transfer Player : Armor Shop (10,11) (Direction: Up)

Setelah itu buat *event* di bawah baris *Else* seperti ini.

- ◆Text : People1(6), Window, Bottom
 - ◊ \n<Pemilik Toko>Kopi macan passwordnya???

Kemudian buat *event Show Choice* seperti ini.



Gambar 1.90. Menambahkan event *Show Choice* untuk masuk ke *Armor Shop*

Setelah itu klik *Show Text* di bawah baris tulisan **When Kopi enak mahal harganya** dan **When Kopi mahal hambar rasanya** dengan tulis seperti ini.

◆Text : People1(6), Window, Bottom

·\n<Pemilik Toko>Salah.

Buat *event* seperti di bawah ini untuk pilihan **When Kopi murah rasa sebanding**

◆Text : People1(6), Window, Bottom

·\n<Pemilik Toko>Silakan masuk.

◆Play SE : Open1 (40, 100, 0)

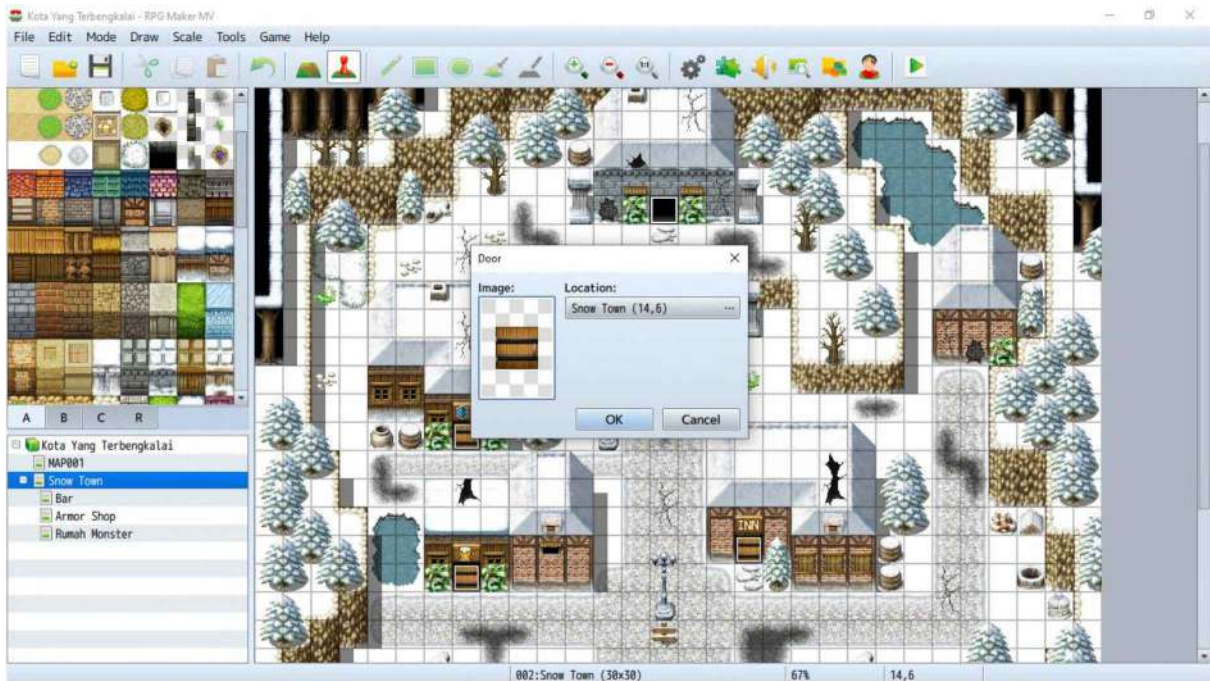
◆Set Movement Route : This Event (Wait)

```

: ◇Turn Left
    ◇Wait: 3 frames
    ◇Turn Right
    ◇Wait: 3 frames
    ◇Turn Up
    ◇Through ON
◆Set Movement Route : Player (Skip, Wait)
    ◇1 Step Forward
◆Play SE : Move1 (40, 100, 0)
◆Transfer Player : Armor Shop (10,11) (Direction: Up)

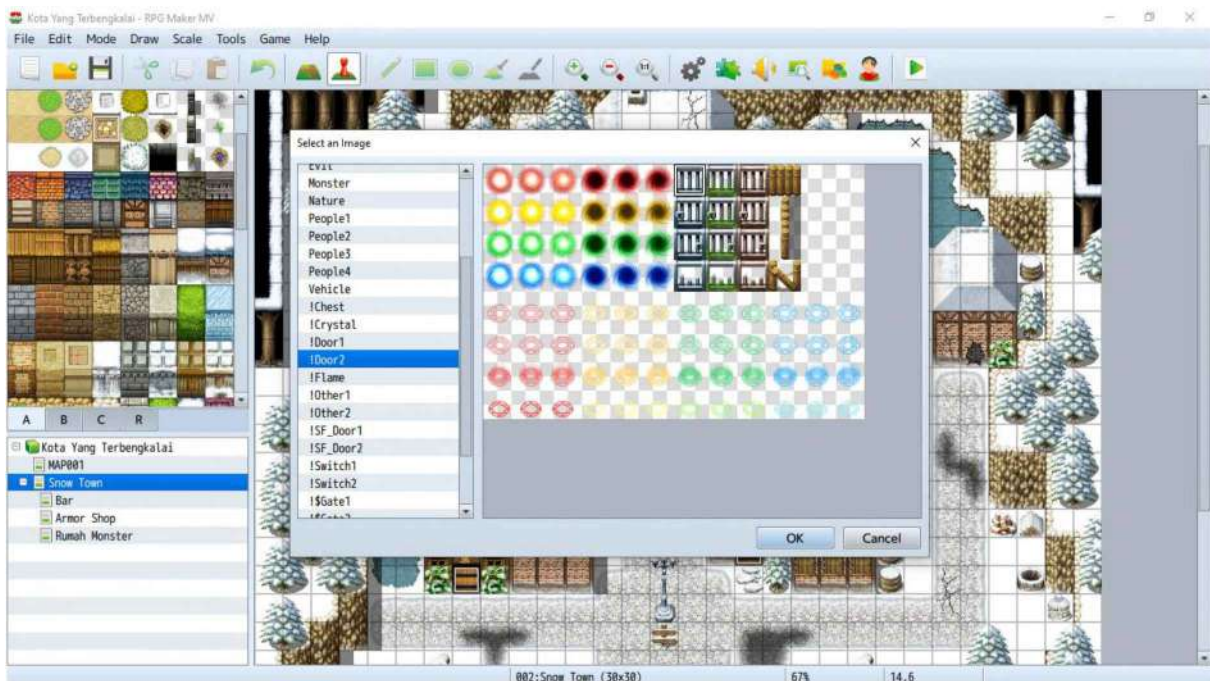
```

Langkah selanjutnya adalah membuat *event* pintu yang terakhir untuk menuju ke *map* Rumah *Monster*. Klik kanan di koordinat (14,6) kemudian pilih *Quick Event Creation* → *Door* atau dengan tombol Ctrl+2.



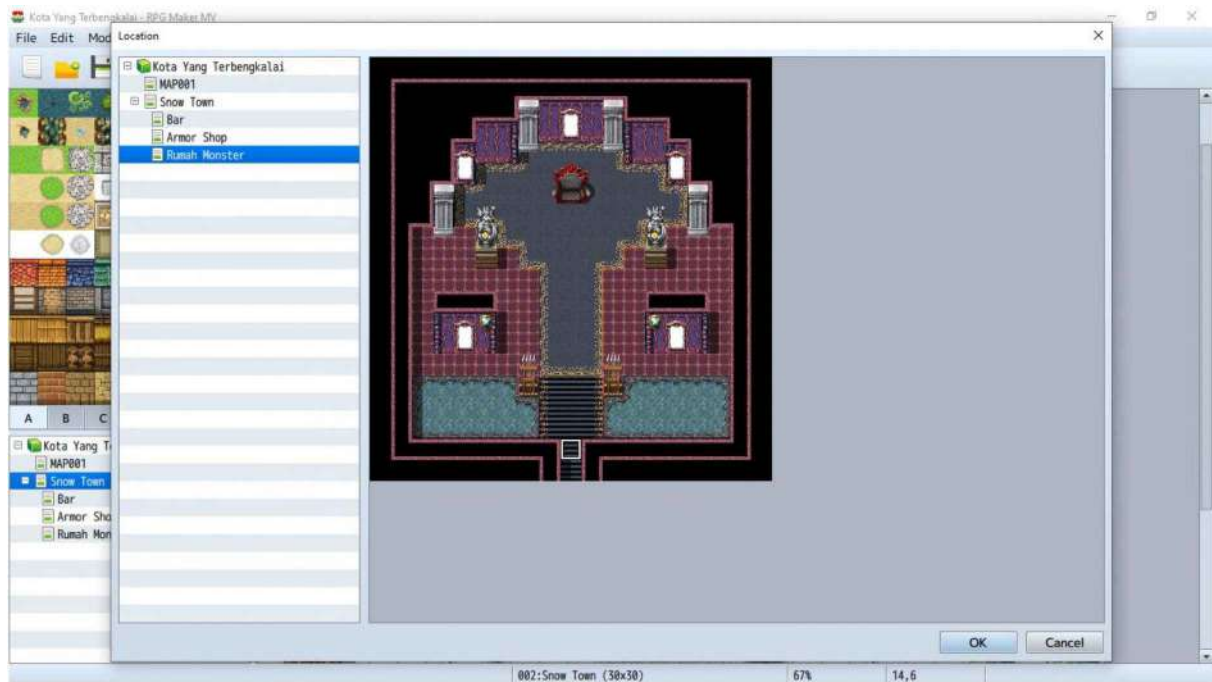
Gambar 3.91. Membuat pintu menuju *map* Rumah *Monster*

Klik dua kali pada gambar pintu lalu pilih *Door2* dan pilih pintu seperti gambar di bawah ini kemudian klik *OK*.



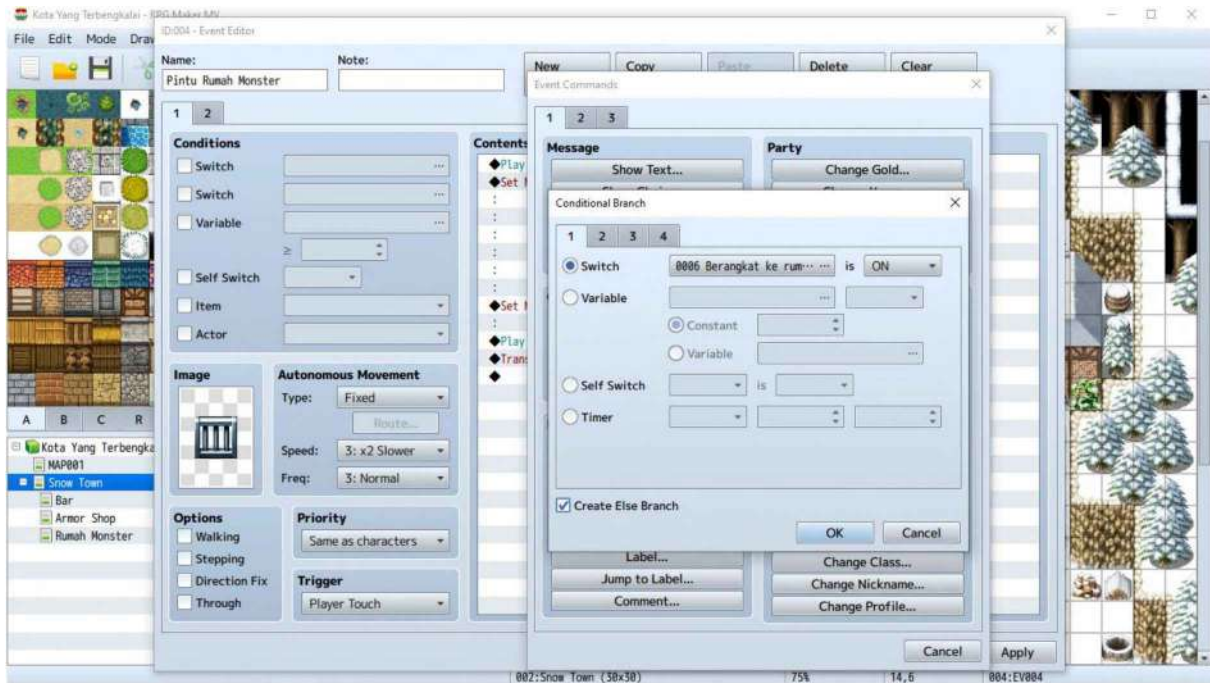
Gambar 3.92. Mengganti gambar pintu Rumah *Monster*

Setelah itu klik tulisan *Snow Town* (14,6) kemudian atur menjadi seperti ini, setelah itu klik *OK*.



Gambar 3.93. Membuat *event transfer map*

Selanjutnya klik dua kali pada gambar pintu Rumah *Monster* lalu beri nama *event* dengan **Pintu Rumah *Monster***. Kemudian klik *Copy Event Page* dan *Paste Event Page* sehingga menjadi dua halaman. Klik dua kali pada baris *Play SE: Open1* (90,100,0) untuk menambahkan *event* di atasnya dan tambahkan *event Conditional Branch* seperti di bawah ini.



Gambar 3.94. Membuat *event Conditional Branch* di pintu Rumah Monster

Setelah itu tambahkan *event* di bawah baris *If*: Pergi ke rumah *Monster* is ON seperti tabel di bawah ini.

◆Text : Actor2(6), Window, Bottom

:\n<Dicky>Semuanya siap?

◆Text : Actor1(1), Window, Bottom

:\n<Kenta>Siap.

◆Text : Actor3(7), Window, Bottom

\n<Izzi>Aduh deg-degan.

◆Text : Actor1(1), Window, Bottom

\n<Kenta>Eh sebelum masuk, sebaiknya pemain kita harus \C[6]save data\C[0] dulu biar aman.

◆Text : Actor3(7), Window, Bottom

:\n<Izzi>Nah betul itu.

◆Text : Actor2(6), Window, Bottom

\n<Dicky>Oke buka \C[6]SAVE MENU\C[0].

Open Save Screen

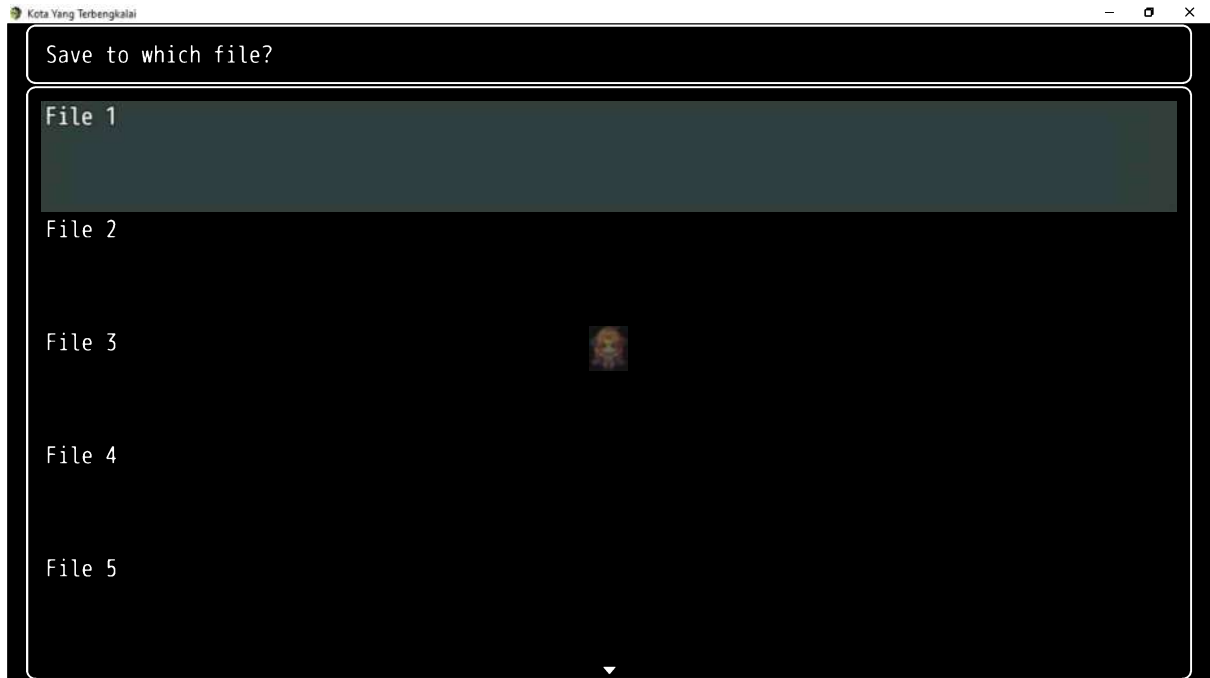
Tabel di atas terdapat beberapa kode yang digunakan, yaitu kode \C[6] dan\C[0]. Dua kode tersebut adalah kode untuk mengubah warna teks yang berasal dari *plugin YEP_MessageCore*. Untuk daftar warna lebih lengkap ada di gambar di bawah ini.



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Gambar 3.95. Kode warna *plugin YEP_MessageCore*

Setelah itu akan muncul menu *save screen* seperti gambar di bawah ini. *Event save screen* dibuat agar pada saat pemain kalah melawan *monster*, pemain tidak perlu mengulangi progres dari awal dan cukup melanjutkan dari progres sebelum memasuki Rumah *Monster*. *Save Menu* juga bisa diakses di *Pause Menu*.



Gambar 3.96. Tampilan *menu save screen*

Setelah membuat *event Open Save Screen*, tambahkan *event* di bawahnya seperti tabel di bawah ini.

- ◆Wait : 60 frames
- ◆Text : Actor3(7), Window, Bottom
: \n<Izzi>Nah gini kan aman.
- ◆Text : Actor1(1), Window, Bottom
: \n<Kenta>Oke kita masuk.
- ◆Text : Actor2(6), Window, Bottom
: \n<Dicky>Satu\|, dua\|, tiga.

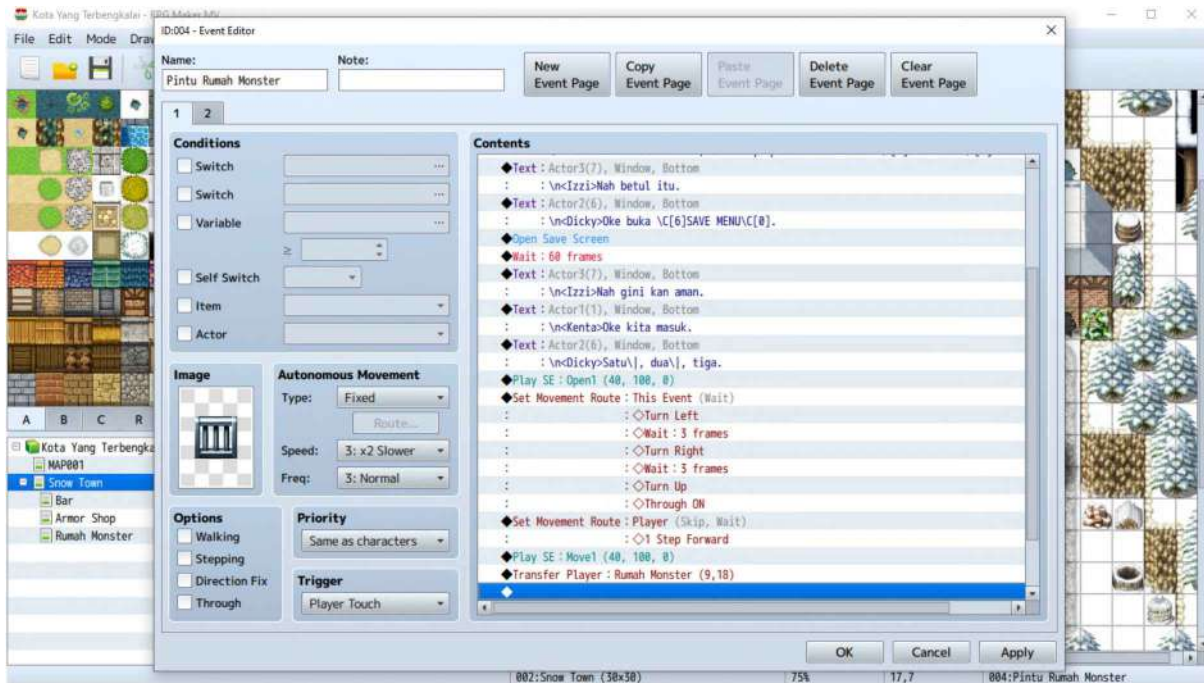
Tabel di atas juga menggunakan kode di *event Show Text*, yaitu `\|`. Kode ini memiliki arti menunggu 1 detik sebelum teks selanjutnya muncul. Sedangkan untuk kode `\.` berarti menunggu $\frac{1}{4}$ detik sebelum teks selanjutnya muncul.

Selanjutnya blok event *Play SE: Open1* (90,100,0) sampai event *Transfer Player: Rumah Monster* (9,18) dengan cara klik pada event *Play SE: Open1* (90,100,0) kemudian tahan tombol *Shift* pada keyboard lalu klik event *Transfer Player: Rumah Monster* (9,18) seperti ini.



Gambar 3.97. Memblok event

Setelah melakukan seperti gambar di atas, klik kanan kemudian pilih *Cut* atau menekan tombol *Ctrl+X* lalu klik kanan pada baris di bawah event *Show Text* karakter Dicky lalu pilih *Paste* atau menekan tombol *Ctrl+V*. Maka hasilnya akan seperti ini.



Gambar 3.98. Memindahkan beberapa *event*

Kemudian tambahkan *event Control Self Switch: A = ON* di bawah *event Transfer Player: Rumah Monster (9,18)*.

Setelah itu tambahkan *event* di bawah baris *Else* lalu klik *OK* setelah semua selesai.

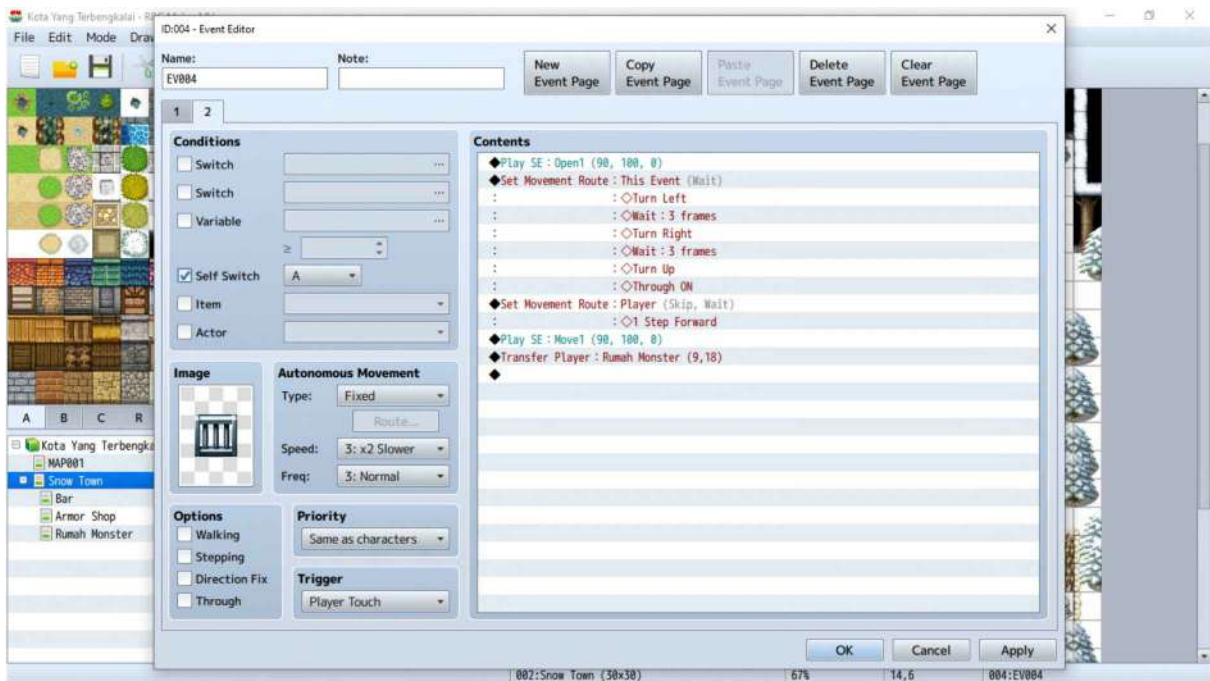
◆**Text** : Actor1(1), Window, Bottom

: \n<Kenta>Yakin mau masuk?

◆**Text** : Actor2(6), Window, Bottom

: \n<Dicky>Nanti aja, kita nggak tau di dalam ada apa.

Selanjutnya klik halaman dua lalu centang tulisan *Self Switch* di kotak *Conditions* yang kemudian akan muncul kotak A setelah itu klik *OK*.



Gambar 3.99. Menambah *Control Self Switch A* di event pintu

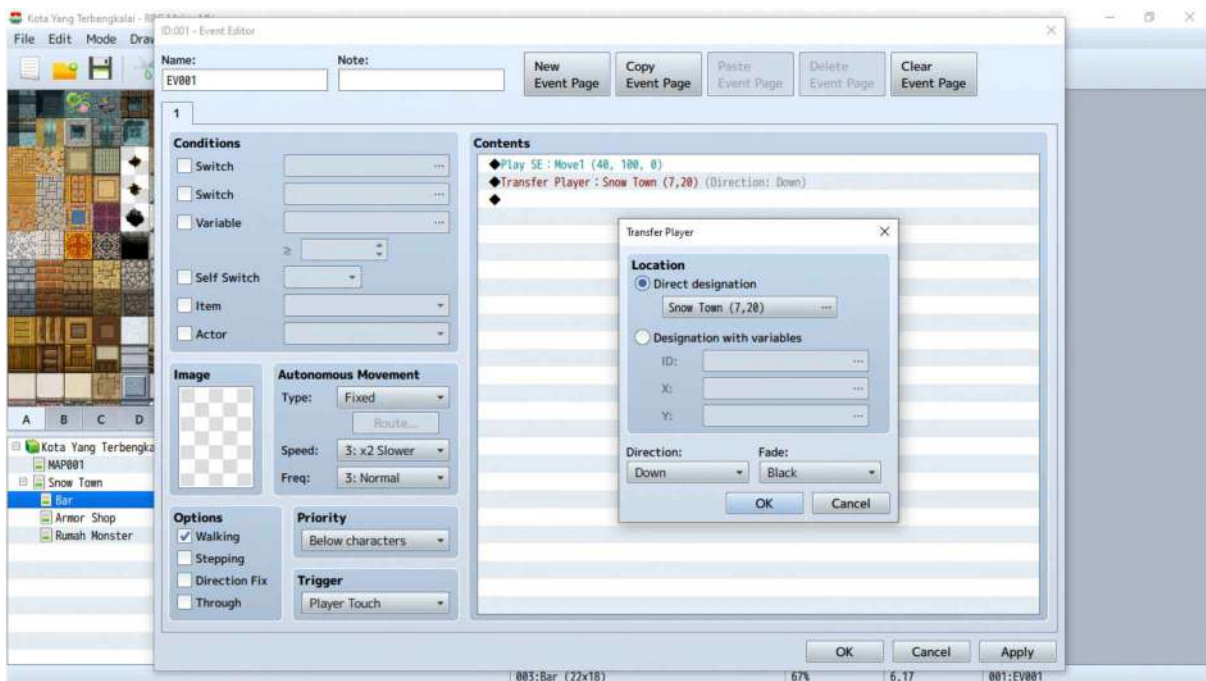
Control Self Switch berfungsi untuk menghindari pengulangan *event*. Pada *event* ini, pemicunya adalah *event* di halaman satu yang apabila telah diatur nanti tidak akan mengalami pengulangan *event* di halaman satu, namun berpindah atau *switch* ke halaman dua untuk pengulangan *event* pintu Rumah Monster selanjutnya.

Membuat Event Keluar Map

Karakter di RPG Maker tidak bisa kembali dari *map* B ke A hanya dengan *event* pintu saja, namun harus terdapat *event* yang akan mentransfer pemain kembali dari *map* B ke A.

Bar

Klik *map Bar* di sebelah kiri sampai *map* utama berganti tampilan menjadi *map Bar*. Setelah itu klik dua kali di koordinat (6,17) sampai muncul jendela *Event Editor*. Setelah jendela *Event Editor* muncul, atur *Triggernya* menjadi *Player Touch* di kolom bagian bawah. Kemudian klik pada baris *Contents* lalu pilih *Play SE → Move1, volume 40%*. Setelah itu klik baris di bawahnya lalu pilih *Transfer Player* dan atur seperti ini lalu klik *OK*.

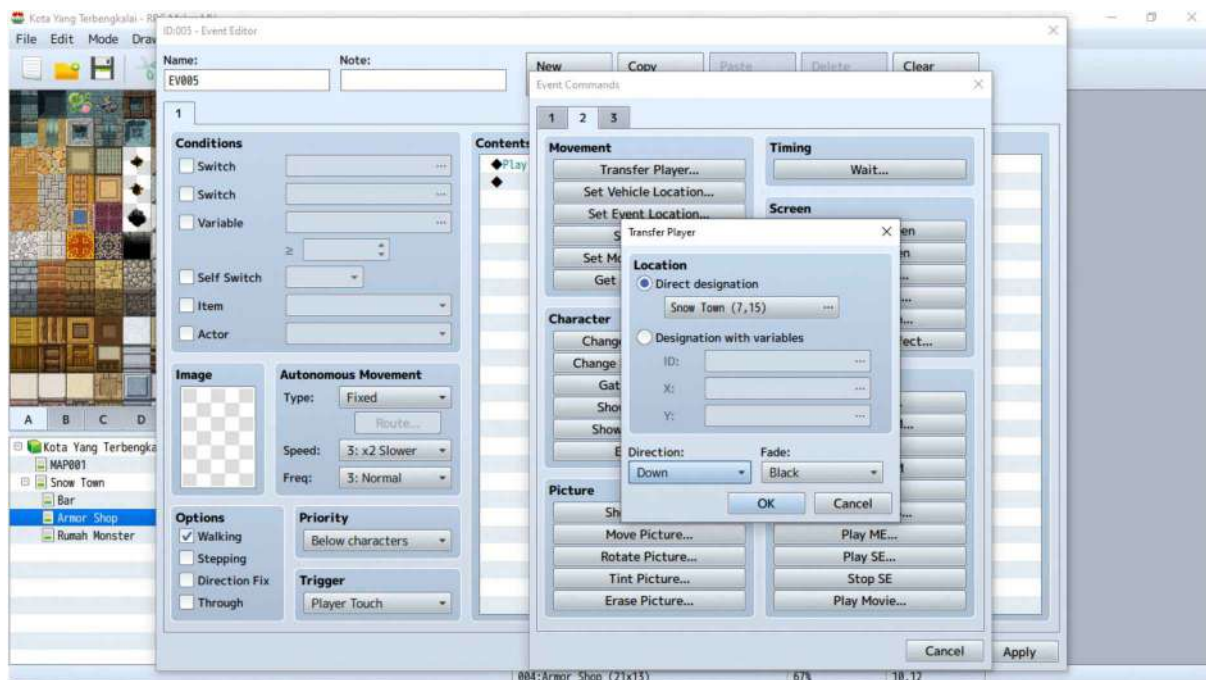


Gambar 3.100. Membuat *event Transfer Player* keluar dari *Bar*

Setelah itu klik baris di bawahnya dan pilih *Fadeout BGM* dan atur durasinya menjadi 3 detik. Klik *OK* setelah semua selesai.

Armor Shop

Selanjutnya adalah mengatur transisi *player* di *map Armor Shop*. Setelah klik *map Armor Shop*, klik dua kali di koordinat (10,12). Atur *Triggernya* menjadi *Player Touch* di kolom bagian bawah. Kemudian klik pada baris *Contents* lalu pilih *Play SE* → *Move1*, *volume 40%*. Setelah itu klik baris di bawahnya lalu pilih *Transfer Player* dan atur seperti ini kemudian klik *OK*.

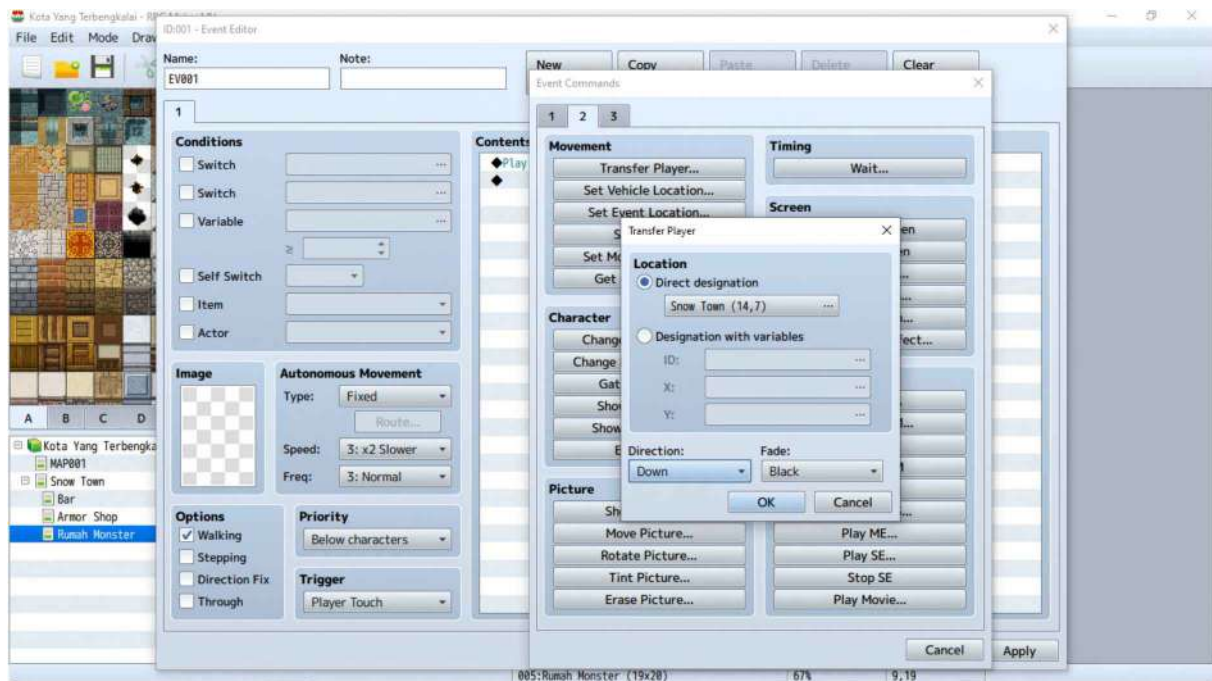


Gambar 3.101. Membuat *event Transfer Player* keluar dari *Armor Shop*

Rumah Monster

Yang terakhir adalah mengatur transisi *player* di *map Rumah Monster*. Setelah klik *map Rumah Monster*, klik dua kali di koordinat (9, 19). Atur *Triggernya* menjadi *Player Touch* di kolom bagian bawah. Kemudian klik pada baris *Contents* lalu pilih *Play SE* → *Move1*, *volume 40%*. Setelah itu klik baris di

bawahnya lalu pilih *Transfer Player* dan atur seperti ini kemudian klik *OK*.



Gambar 3.102. Membuat *event Transfer Player* keluar dari *Rumah Monster*

Setelah itu klik baris di bawahnya dan pilih *Fadeout BGM* dan atur durasinya menjadi 3 detik. Klik *OK* setelah semua selesai.

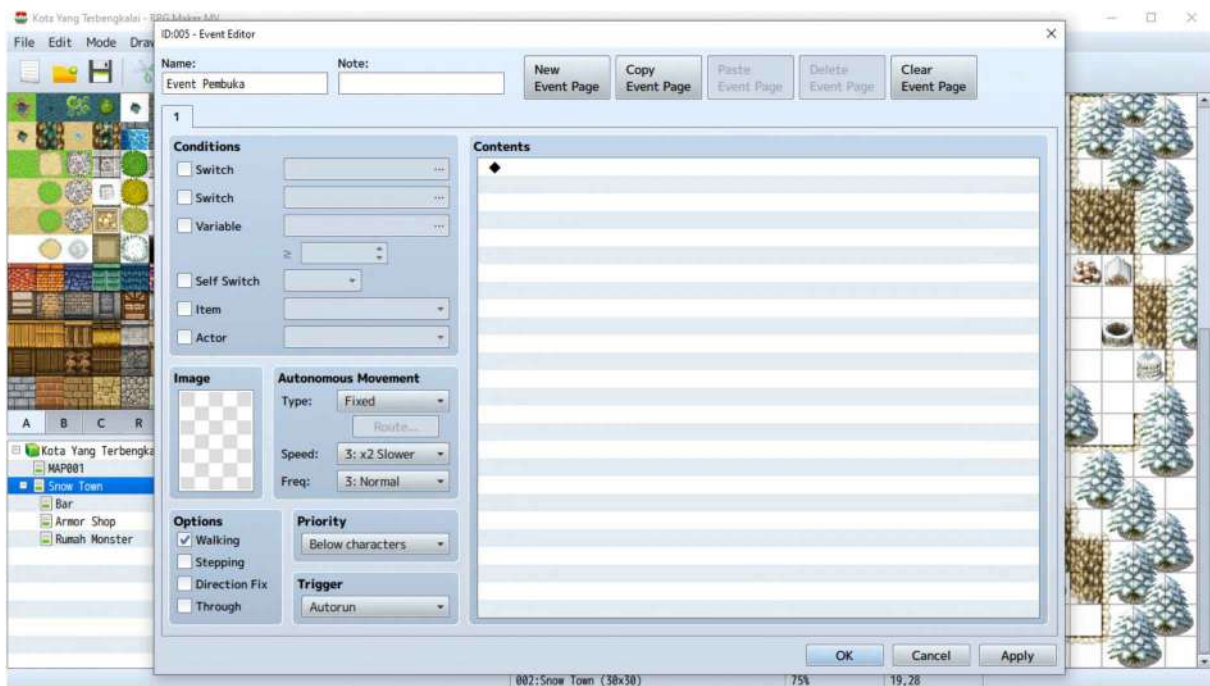
Membuat Event Utama

Event Pembuka

Langkah selanjutnya adalah membuat *event* utama yang berfungsi sebagai misi utama pemain untuk bermain. Cara membuat *event* ini adalah dengan klik dua kali di tempat yang

tidak bisa disentuh pemain di *map Snow Town*, sebagai contoh saya menggunakan koordinat (19,28).

Setelah jendela *Event Editor* muncul, beri nama *eventnya* dengan nama **Event Pembuka**. Atur *Triggernya* menjadi *Autorun* untuk halaman satu di kolom bagian bawah seperti gambar di bawah ini lalu klik *OK*.



Gambar 3.103. Membuat *Event Pembuka*

Selanjutnya klik dua kali pada baris kotak *Contents* dan buat *event* seperti di bawah ini.

◆Control Self Switch: A = ON

◆Fadein Screen

◆Wait :60 frames

◆Text :Actor3(7), Window, Bottom

: \n<Izzi>Jadi ini\.\., yang namanya Atlantis.

◆Text :Actor1(1), Window, Bottom

: \n<Kenta>Ngawur. Atlantis itu di bawah laut.

◆Text :Actor2(6), Window, Bottom

: \n<Dicky>Betul\.\., kota itu juga kota fiksi jadi nggak ada di dunia nyata.

◆Text :Actor3(7), Window, Bottom

: \n<Izzi>Bukannya kita juga karakter fiksi?

◆Text :Actor2(6), Window, Bottom

: \n<Dicky>Hmm\..\..\| Kalau kita karakter fiksi\.\., aku penasaran siapa yang menggerakkan
` kita ke tempat ini.

◆Text :Actor3(7), Window, Bottom

: \n<Izzi>Kenapa milihnya tempat ini? \.\.apa nggak ada yang lebih bagus?

◆Text :Actor1(1), Window, Bottom

: \n<Kenta>Kayaknya yang buat game udah dikejar deadline\.\., makanya tempatnya kayak gini.

◆Text :Actor2(6), Window, Bottom

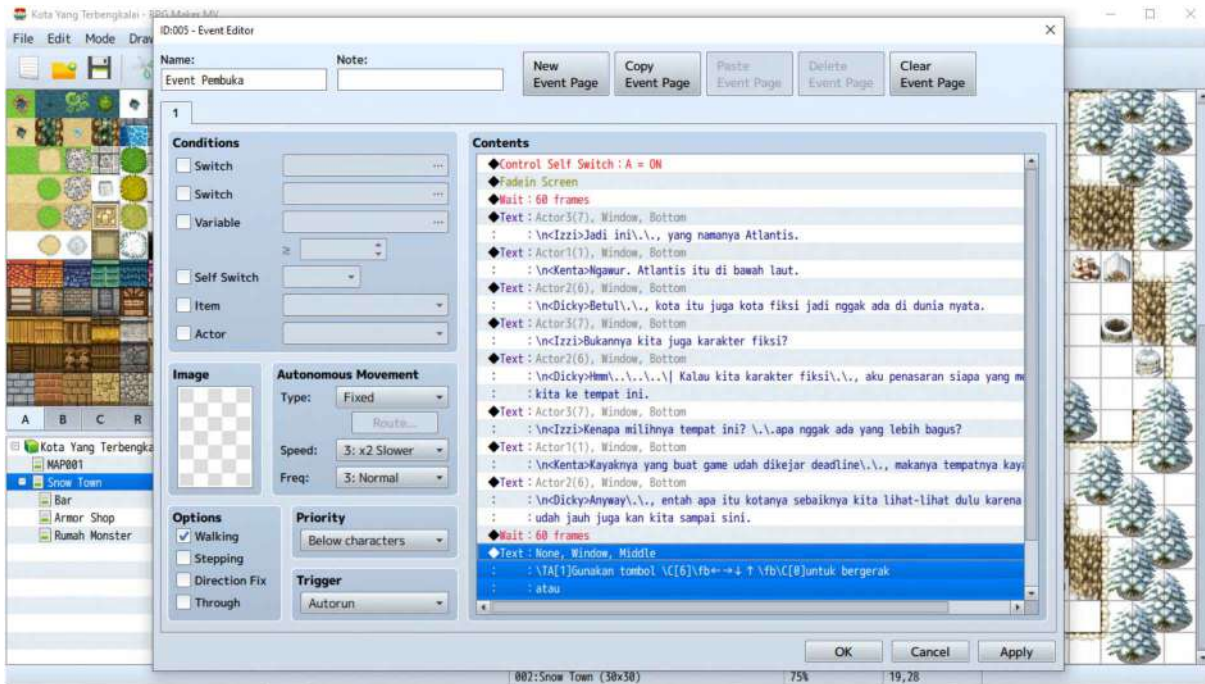
: \n<Dicky>Anyway\.\., entah apa itu kotanya sebaiknya kita lihat-lihat dulu karena
` udah jauh juga kan kita sampai sini.

◆Wait :60 frames

Setelah percakapan tersebut selesai, pemain bisa langsung menggerakkan karakter untuk menjalankan misi. Agar tidak membuat pemain bingung, setelah *event Wait: 60 frames* sebaiknya diberikan petunjuk *control movement* pemain dengan menambahkan *event Show Text* seperti di bawah ini

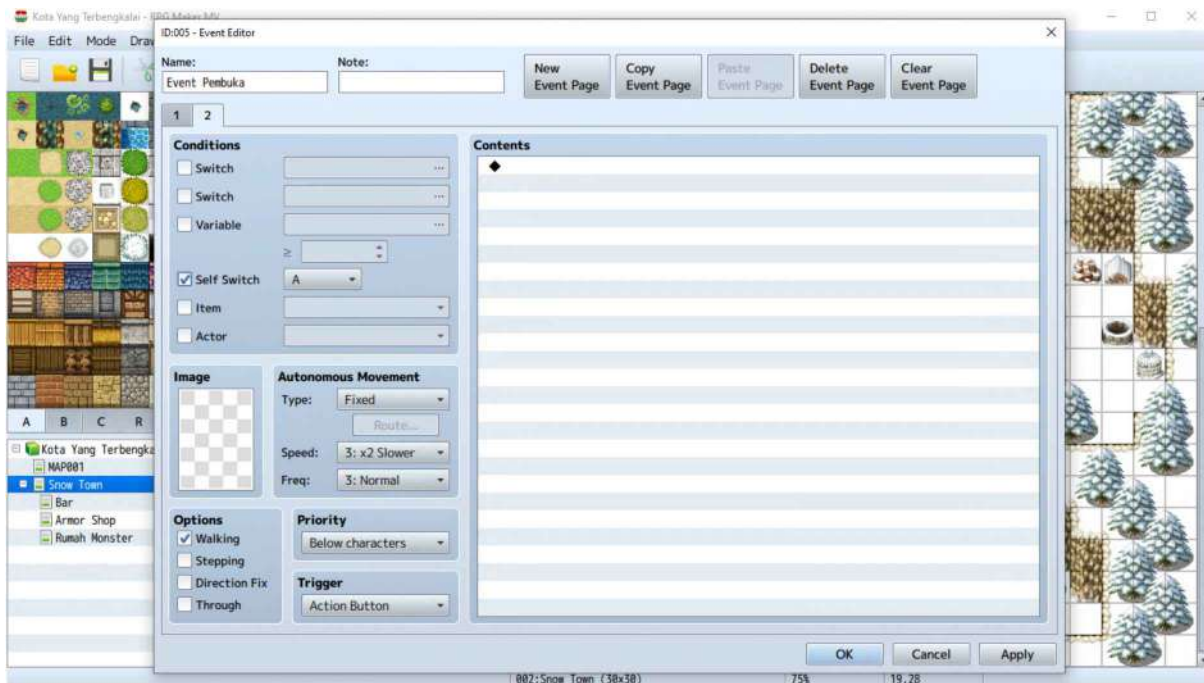
◆Text :None, Window, Middle

- \TA[1]Gunakan tombol \C[6]\fb←→↓↑\fb\C[0] untuk bergerak
- atau
- tombol \C[6]\fbWASD\fb\C[0] jika sudah diganti di tombol menu.
- Gunakan tombol \C[6]\fbENTER\fb\C[0] atau \C[6]\fbSPACE\fb\C[0] untuk berinteraksi



Gambar 3.104. Tampilan halaman satu *Event Pembuka* pada *Event Editor*

Selanjutnya klik *New Event Page* untuk membuat halaman dua di *Event Editor*. Kemudian centang pada *Self Switch A* pada kolom kiri halaman dua dan biarkan kotak *Contents* kosong lalu klik *OK*.

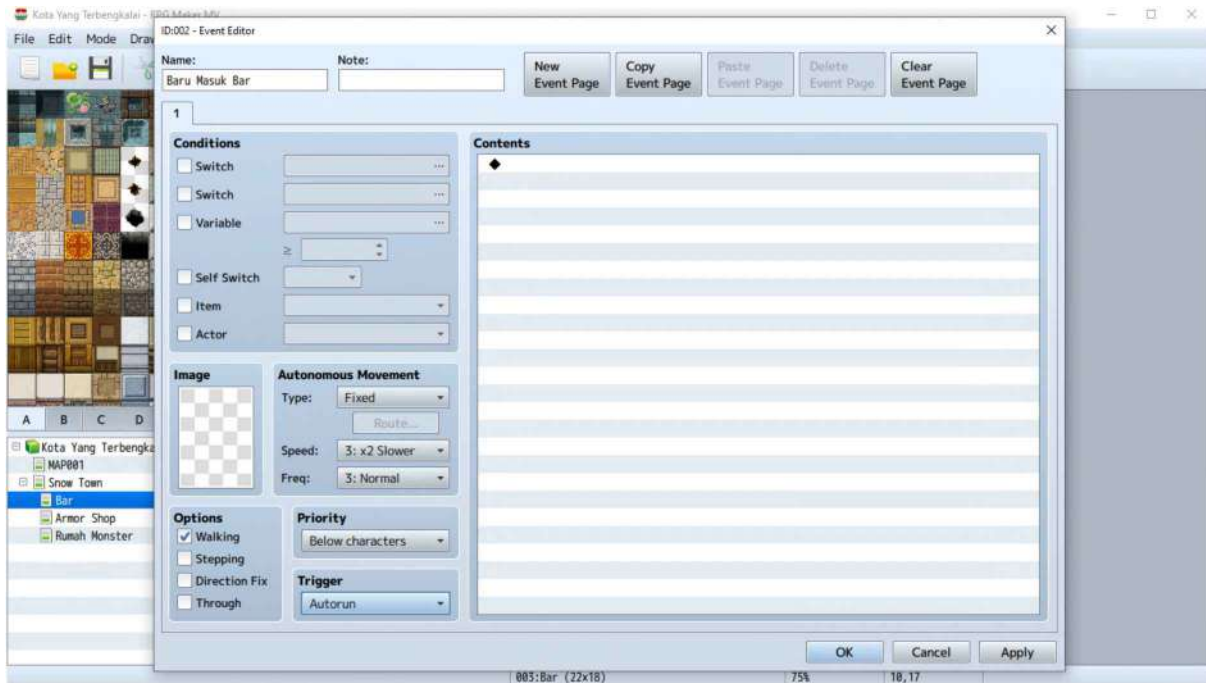


Gambar 3.105. Tampilan halaman dua *Event* Pembuka pada *Event Editor*

Masuk Bar

Event selanjutnya yang dibuat adalah *event* di *map Bar*. Klik *map Bar* di sebelah kiri sampai *map* berganti tampilan menjadi *map Bar*.

Setelah itu klik dua kali di tempat yang tidak bisa disentuh pemain. Sebagai contoh saya menggunakan koordinat (10,17). Setelah jendela *Event Editor* muncul, beri nama *event*-nya dengan nama **Baru Masuk Bar**. Atur *Triggernya* menjadi *Autorun* untuk halaman satu di kolom bagian bawah seperti gambar di bawah ini lalu klik *OK*.



Gambar 3.106. Tampilan halaman satu *Event* Baru Masuk *Bar* pada *Event Editor*

Selanjutnya klik dua kali pada baris di kotak *Contents* dan buat *event* seperti di bawah ini.

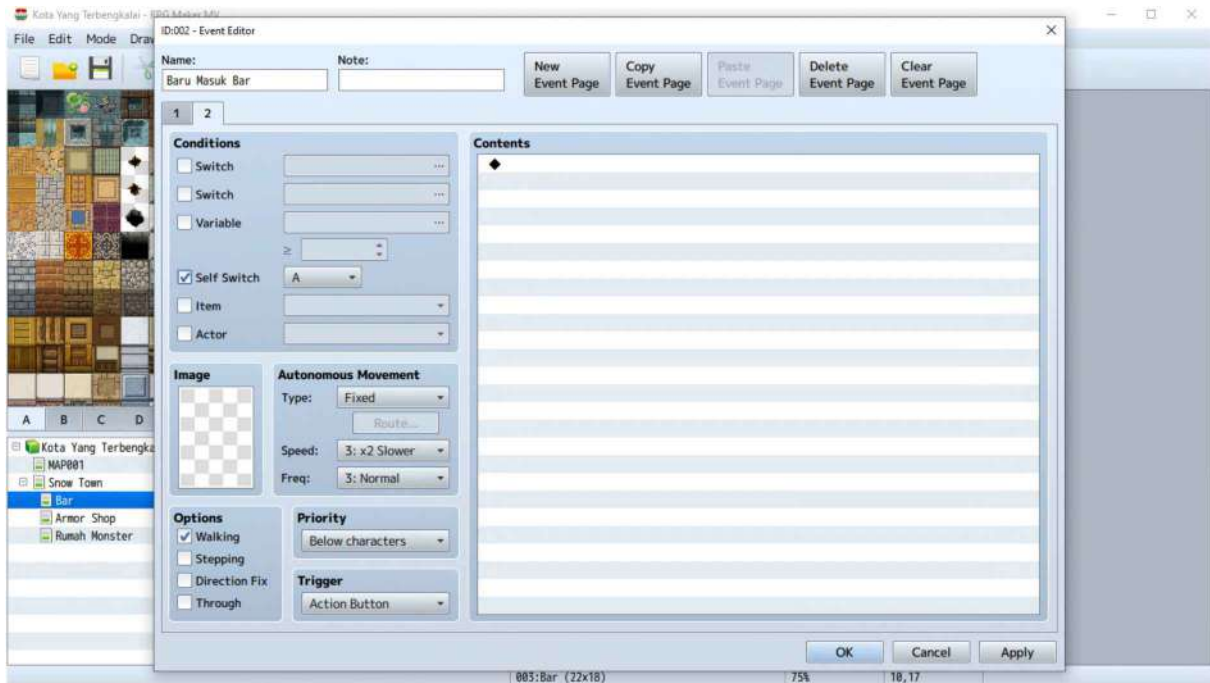
◆Control Self Switch: A = ON

◆Wait :60 frames

◆Text :Actor1(1), Window, Bottom

: \n<Kenta>Akhirnya ada orang juga.

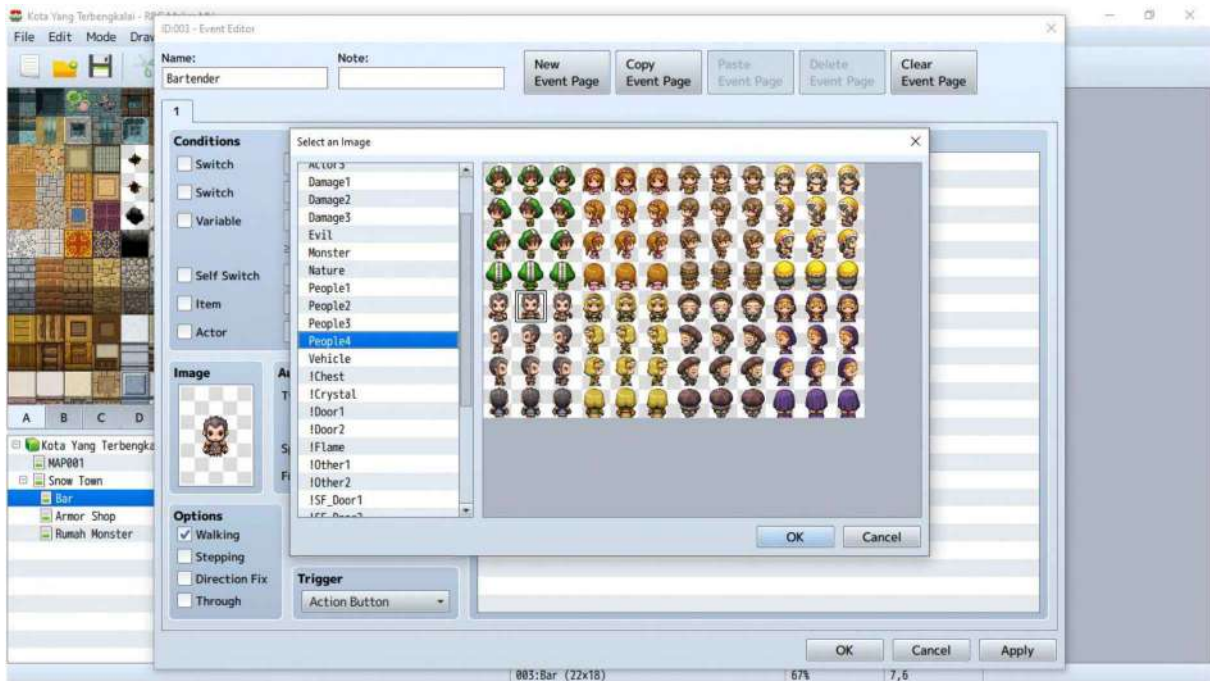
Setelah itu klik *New Event Page* untuk membuat halaman dua di *Event Editor*. Kemudian centang pada *Self Switch A* pada kolom kiri halaman dua dan biarkan kotak *Contents* kosong lalu klik *OK*.



Gambar 3.107. Tampilan halaman dua *Event* Baru Masuk *Bar* pada *Event Editor*

Bartender

Langkah selanjutnya adalah menambahkan *NPC (Non Playable Character)*, yaitu Bartender yang akan melakukan percakapan dengan karakter pemain. Langkah pertama adalah klik dua kali di koordinat (7,6) kemudian klik dua kali di kolom *image* dan pilih *People4(4)* lalu klik *OK*.



Gambar 3.108. Membuat *NPC* Bartender

Kemudian klik *Copy Event Page* dan *Paste Event Page* untuk menduplikat halaman menjadi dua lalu namai *event* ini **Bartender**. Setelah itu tambahkan *event* di kotak *Contents* halaman satu seperti tabel di bawah ini.

◆Text :People4(4), Window, Bottom

:\n<Bartender>Sepertinya kalian bertiga terlihat asing.

◆Text :Actor1(1), Window, Bottom

:\n<Kenta>Kami cuma bocil yang tersesat pak.

◆Text :Actor3(7), Window, Bottom

:\n<Izzi>Kenapa tempat ini seperti kapal pecah?

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>To the point sekali ya.

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Dulu ada sebuah monster yang menyerang kota ini.

◆Text :Actor3(7), Window, Bottom

:\\n<Izzi>Apa monster itu masih di sini?

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Iya\\|. Sekarang monster itu berada di rumah di ujung kota ini\\|.

:Makanya kalian akan menemukan tulisan larangan untuk ke sana.

◆Text :Actor1(1), Window, Bottom

:\\n<Kenta>Apakah monster itu akan menyerang lagi?

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Aku tidak tau\\|. Kami yang tersisa di sini tidak pernah keluar dari sini

:semenjak kejadian hari itu.

◆Text :Actor2(6), Window, Bottom

:\\n<Dicky>Apa tidak takut kalau monster itu datang ke sini?

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Kemungkinannya kecil\\|. Kalian lihat pohon kacang yang ada di pintu depan?\\.\\.

:Itu adalah pencegahan monster itu untuk ke sini.

◆Text :Actor3(7), Window, Bottom

:\\n<Izzi>Kalau gitu kenapa enggak lempar aja semua kacang ke monsternya?

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Kacang di sini terbatas\\., itupun efeknya hanya sementara.

◆Wait :120 frames

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Tunggu sebentar.

◆Wait :60 frames

◆Text :People4(4), Window, Bottom

:\\n<Bartender>Apa salah satu dari kalian ada yang bernama Dick?

◆Text :Actor2(6), Window, Bottom

: \n<Dicky> Saya pak, Dicky.

◆Text : Actor2(6), Window, Bottom

: \n<Dicky> Kok tau nama saya?

◆Text : People4(4), Window, Bottom

: \n<Bartender> Apa jangan-jangan ramalan itu benar?

◆Text : Actor3(7), Window, Bottom

: \n<Izzi> Ramalan apa???

◆Text : People4(4), Window, Bottom

: \n<Bartender> Dulu ada ramalan yang mengatakan suatu saat ada sekelompok orang yang akan

: mengalahkan monster itu. \|\|

: Aku cuma tidak percaya kalau ternyata sekelompok bocil.

◆Text : Actor3(7), Window, Bottom

: \n<Izzi> HEY\., JANGAN MEREMEHKAN KAMI\|. Kami itu BBB\., Bukan Bocil Biasa\|. Buktinya kami bisa

: pergi dari rumah sampai jauh ke sini.

◆Text : Actor1(1), Window, Bottom

: \n<Kenta> Itu karena kita bodoh, Izzi\|. Bocil seumuran kita mana ada yang main sampai sini\|.

: Lihat aja waktu kita nanti sampai rumah, orangtua kita udah menyiapkan sapu

: untuk memukul kita.

◆Text : Actor3(7), Window, Bottom

: \n<Izzi> KAMU BILANG AKU BODOH?

◆Text : Actor2(6), Window, Bottom

: \n<Dicky> Hei cukup\|. Pak tua, kalau memang ramalan itu benar,\.\. bagaimana caranya untuk

: mewujudkannya?

◆Text : Actor1(1), Window, Bottom

: \n<Kenta> Yang bener aja, Dick. Kalau kita mati gimana?

◆Text : Actor2(6), Window, Bottom

:\n<Dicky>Tenang. Kalau kita mati salahin yang mainin kita soalnya mainnya nggak bener.

◆Text : **People4(4), Window, Bottom**

:\n<Bartender>Kalian bisa bertanya ke bandar yang ada di sana untuk informasi lebih lanjut\|.

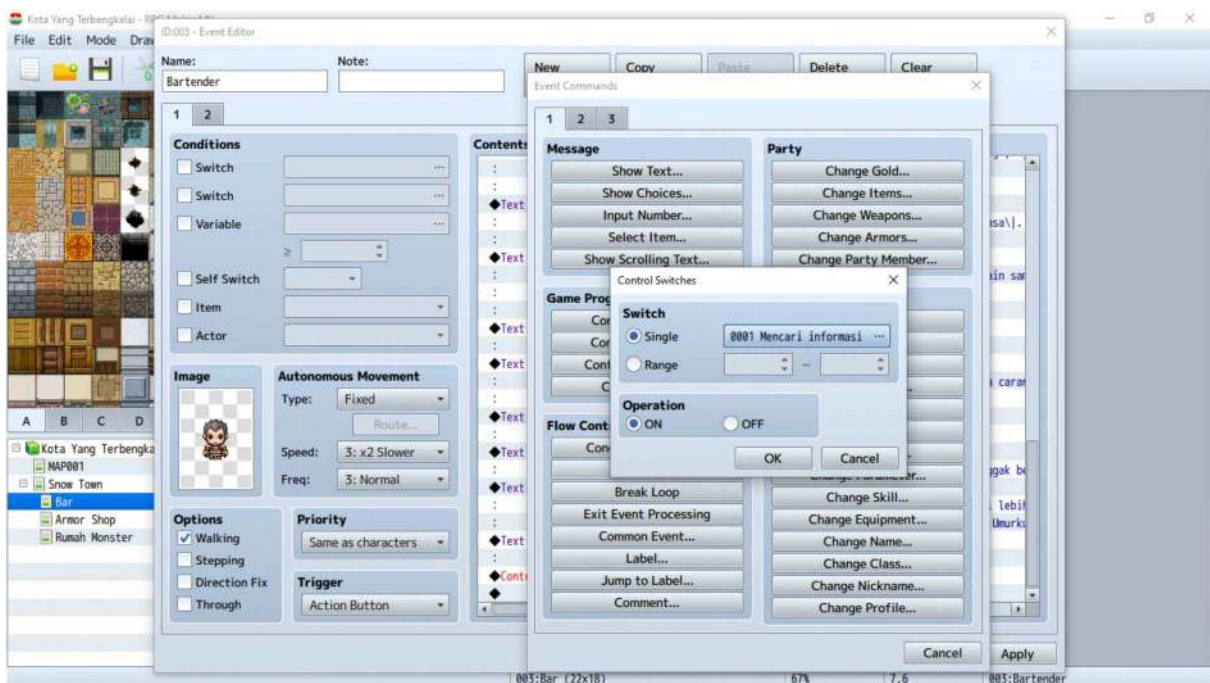
:O ya satu lagi\.\., jangan panggil aku Pak Tua\.\., panggil Paman saja\.\.. Umurku masih 40an.

◆Text : **Actor3(7), Window, Bottom**

:\n<Izzi>Siap, Paman\.\.. Semoga harimu menyenangkan.

Control Self Switch A = ON

Control Switches: #0001 Mencari informasi = ON



Gambar 3.109. Membuat *event Control Switches*

Setelah *Control Switches #0001* diaktifkan maka menandakan kalau percakapan Bartender dengan karakter pemain telah selesai dan jika berbicara kepada Bartender lagi maka akan muncul percakapan berbeda yang akan ditulis di *Event Editor* halaman dua.

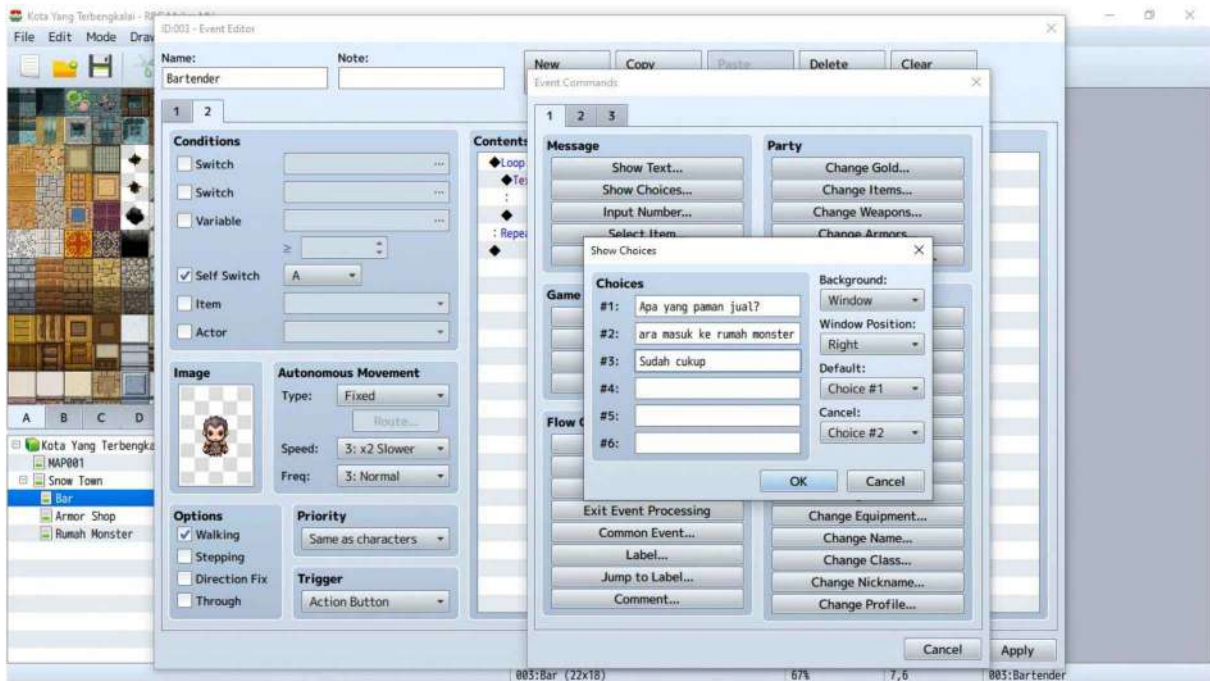
Gambar 3.110. Membuat *event Loop*

Setelah muncul tampilan seperti di atas, kemudian klik dua kali pada baris di bawah *Loop* dan pilih *Show Text* seperti di bawah ini.

◆Text : People4(4), Window, Bottom

:\n<Bartender>Ada yang bisa kubantu?

Selanjutnya pilih *Show Choice* dan masukkan pilihan seperti di bawah ini.



Gambar 3.111. Membuat *event Show Choice* Bartender

Selanjutnya tulis seperti ini di bawah pilihan “Apa yang paman jual?”

◆Text : **People4(4), Window, Bottom**

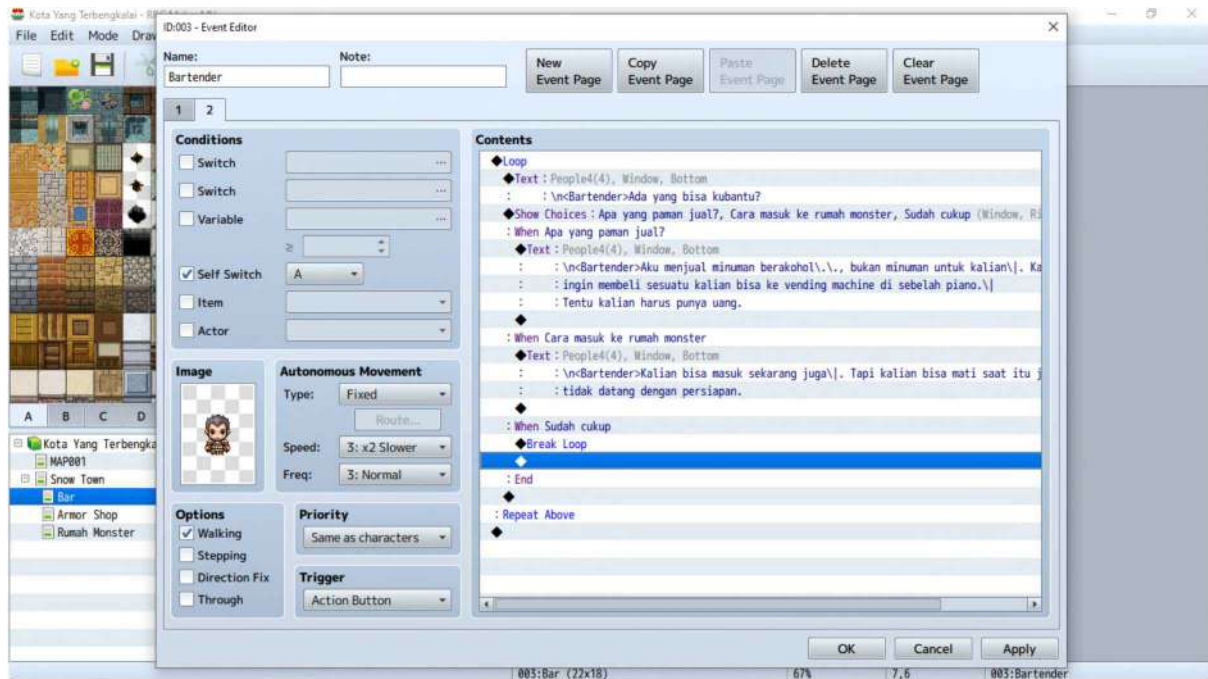
·\n<Bartender>Aku menjual minuman berakohol\., bukan minuman untuk kalian\|. Kalau kalian
·ingin membeli sesuatu kalian bisa ke vending machine di sebelah piano.\|
·Tentu kalian harus punya uang.

Kemudian tulis seperti ini di bawah pilihan “Cara masuk ke rumah monster”

◆Text : **People4(4), Window, Bottom**

·\n<Bartender>Kalian bisa masuk sekarang juga\|. Tapi kalian bisa mati saat itu juga kalau kalian
·tidak datang dengan persiapan.

Untuk pilihan terakhir (Sudah cukup), pilih *event Break Loop* seperti di bawah ini.

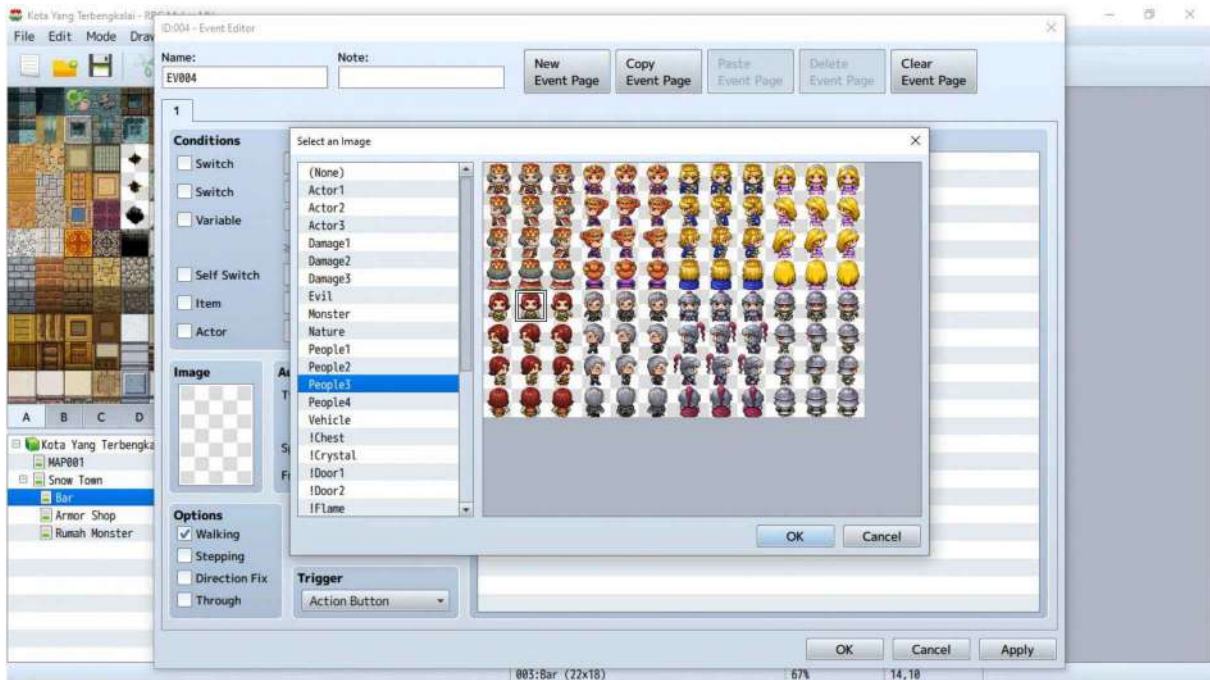


Gambar 3.112. Membuat *event Break Loop*

Event Break Loop berfungsi untuk keluar dari *event Loop*. Karena saat pemain memilih pilihan satu atau dua, pemain akan kembali ditampilkan *event Show Choice* sebelum akhirnya berhenti muncul setelah memilih pilihan ketiga. Klik *OK* setelah selesai membuat seperti di atas.

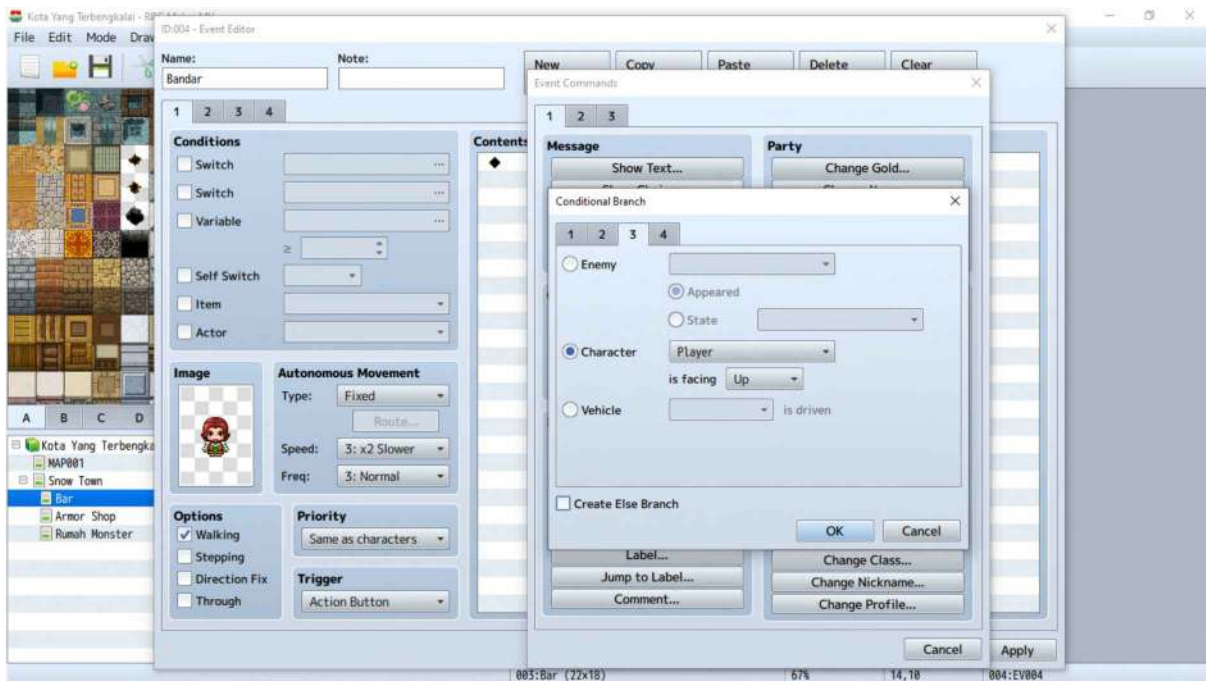
Bandar

Setelah membuat *event* Bartender selesai, selanjutnya buat *event* Bandar. Klik dua kali di koordinat (14,10) lalu klik dua kali pada kota *image* kemudian pilih seperti gambar di bawah ini. Setelah itu klik *OK*.



Gambar 3.113. Membuat *event* Bandar

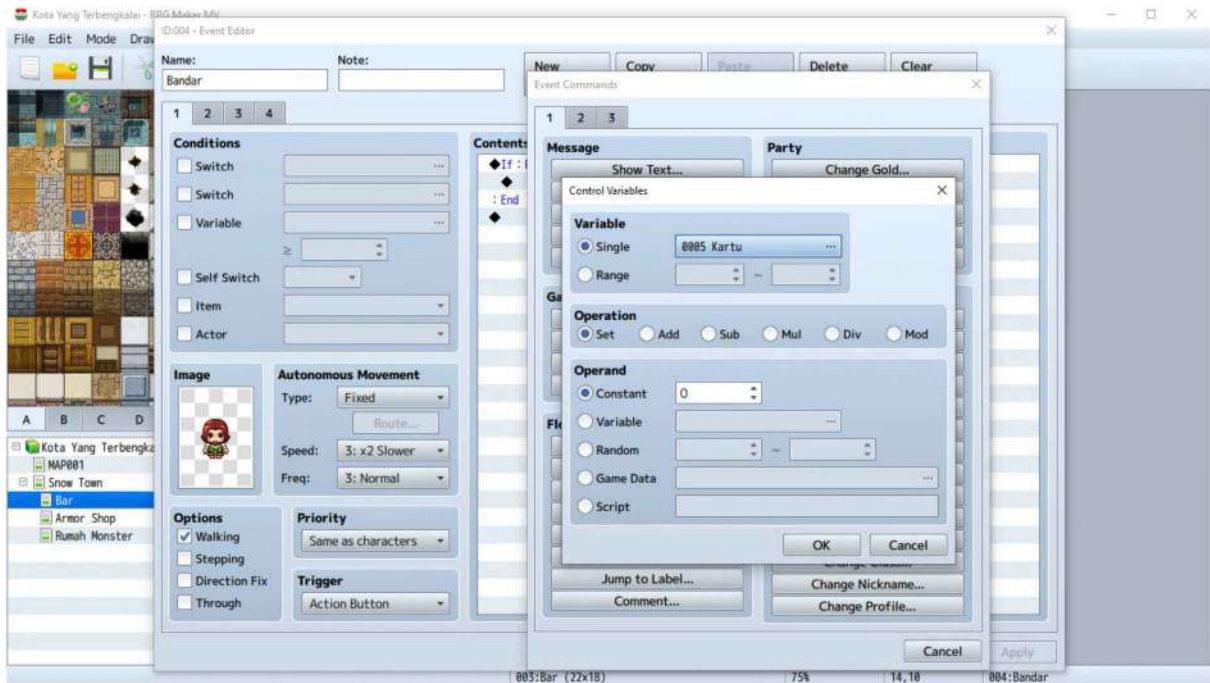
Selanjutnya namai *event* ini **Bandar** lalu klik baris pada kotak *Contents* lalu pilih *Conditional Branch* seperti gambar di bawah ini.



Gambar 3.114. Membuat *event Conditional Branch*

Kemudian pilih *Character* → *Player is facing up* lalu klik *OK*. Ini adalah kondisi bahwa untuk berinteraksi dengan *NPC Bandar*, pemain harus menghadap ke atas dan tidak bisa berinteraksi dari samping atau atas *Bandar*.

Setelah itu klik *Copy Event Page* dan klik *Paste Event Page* tiga kali untuk menduplikat halaman menjadi empat. Kemudian klik dua kali pada baris di bawah *If* halaman satu lalu klik pilih *Control Variable* kemudian pilih *Variable 0005 Kartu* seperti gambar di bawah.



Gambar 3.115. Membuat *Variable* Kartu

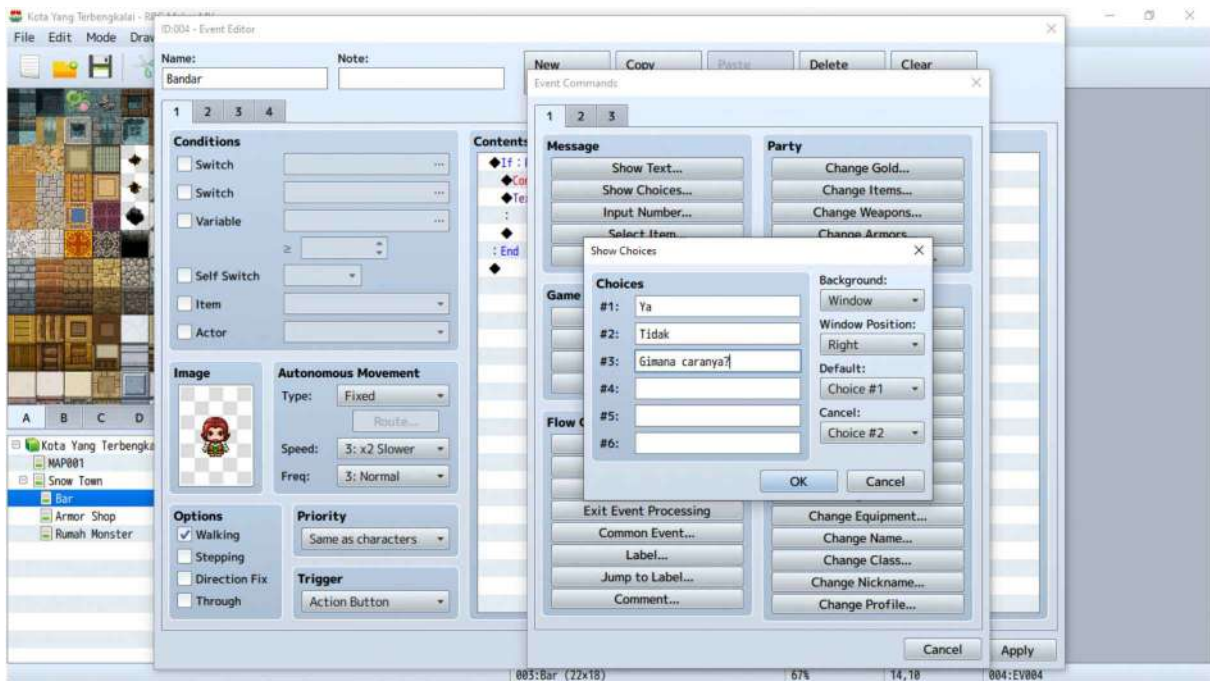
Kemudian lakukan seperti gambar di atas kemudian klik *OK*. *Control Variable* ini akan mengatur *Variable* Kartu menjadi = 0 yang akan berfungsi sebagai reset variabel Kartu pemain.

Selanjutnya klik dua kali di bawah *Control Variables* #0005 Kartu = 0 dan pilih *Show Text* lalu tulis seperti di bawah ini.

◆Text : People3(4), Window, Bottom

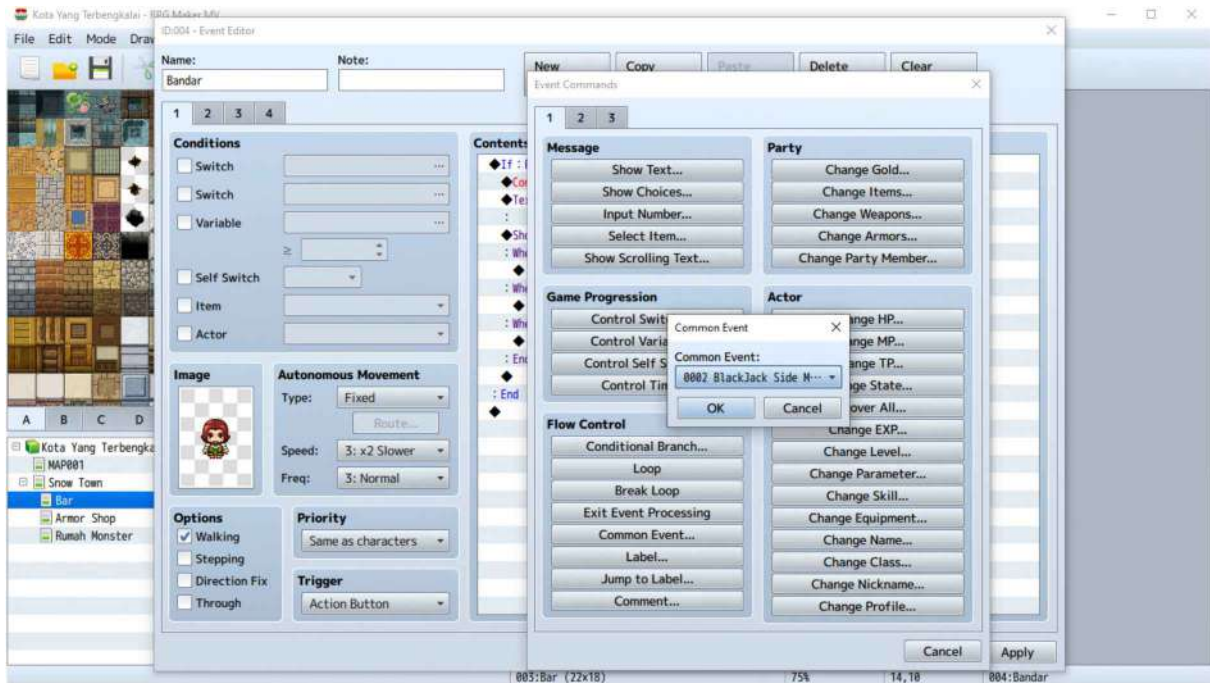
:\n<Bandar>Halo nak, mau main BlackJack?

Selanjutnya klik di bawah baris *text* di atas dan pilih *Show Choice* seperti gambar di bawah ini.



Gambar 3.116. Membuat *event Show Choice* sebelum bermain BlackJack

Klik baris di bawah baris ***When Ya*** dan pilih *event Common Event*. Kemudian pilih *Common Event: 0002 BlackJack Side Mission* setelah itu klik *OK*.



Gambar 3.117. Menambahkan *Common Event* di *event Show Choice*

Biarkan baris di bawah **When Tidak** kosong lalu klik baris di bawah **When Gimana caranya?** dan tulis seperti di bawah ini.

◆Text :Actor3(7), Window, Bottom

:\n<lzzi>Kami tidak tau caranya bermain.

◆Text :People3(4), Window, Bottom

:\n<Bandar>Oh itu mudah\|. Kalian hanya perlu mempunyai kartu yang jumlahnya 21\.\.,

:lebih dari itu kalian kalah.

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>Boleh, kelihatannya cukup mudah.

◆Text :People3(4), Window, Bottom

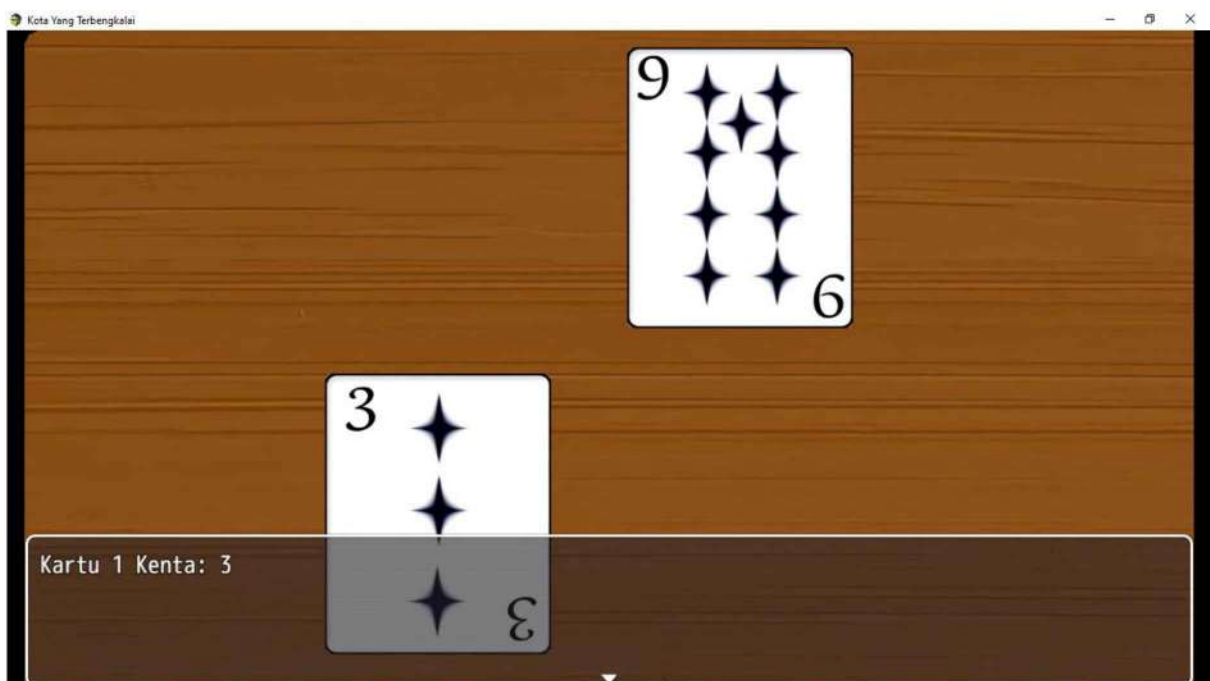
:\n<Bandar>Oke, sekarang siapa yang ingin bermain?

◆Text :People4(4), Window, Bottom

:\n<lzzi>\n<lzzi>Biar pemain kami yang memutuskan\|. Caranya adalah pilih \C[6]FORMATION\C[0] di pause menu\.\.,

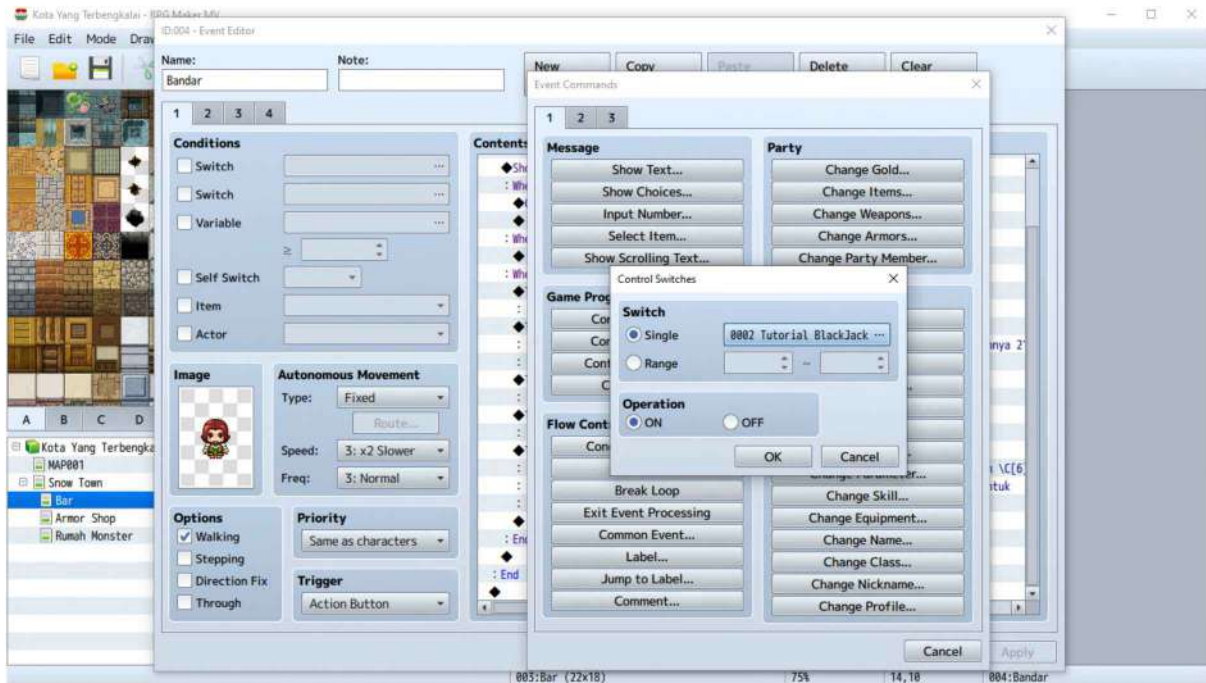
- Setelah itu pilih salah satu dari kami dan letakkan di urutan teratas untuk
- bermain.

Percakapan terakhir menunjukkan bahwa Izzi menginstruksikan pemain untuk memilih karakter mana yang akan bermain BlackJack karena di dalam *Common Event BlackJack Main Mission* dan *Side Mission* terdapat kode \P[1] yang berarti kode tersebut akan memanggil karakter yang berada di urutan terdepan.



Gambar 3.118. Tampilan teks saat aktor Kenta berada di urutan paling depan

Selanjutnya masukkan *event Control Switches* dan pilih seperti gambar di bawah ini.



Gambar 3.119. Membuat *Control Switch* Tutorial *BlackJack*

Control Switch ini bertujuan untuk memberikan variasi percakapan pada saat pemain ingin bermain BlackJack, yaitu saat *Control Switches*: #0001 Mencari informasi = *ON* setelah karakter selesai berbicara kepada Bartender.

Selanjutnya klik halaman dua lalu centang tulisan Switch di kotak *Conditions* kemudian pilih 0001 Mencari informasi. Setelah itu klik baris di bawah baris *If: Player is Facing Up* lalu buat *event* seperti di bawah ini.

◆Text :Actor2(6), Window, Bottom

·\n<Dicky>Halo pak, kudengar anda tau informasi tentang monster itu.

◆Text :People3(4), Window, Bottom

·\n<Bandar>Benar, memangnya kenapa?

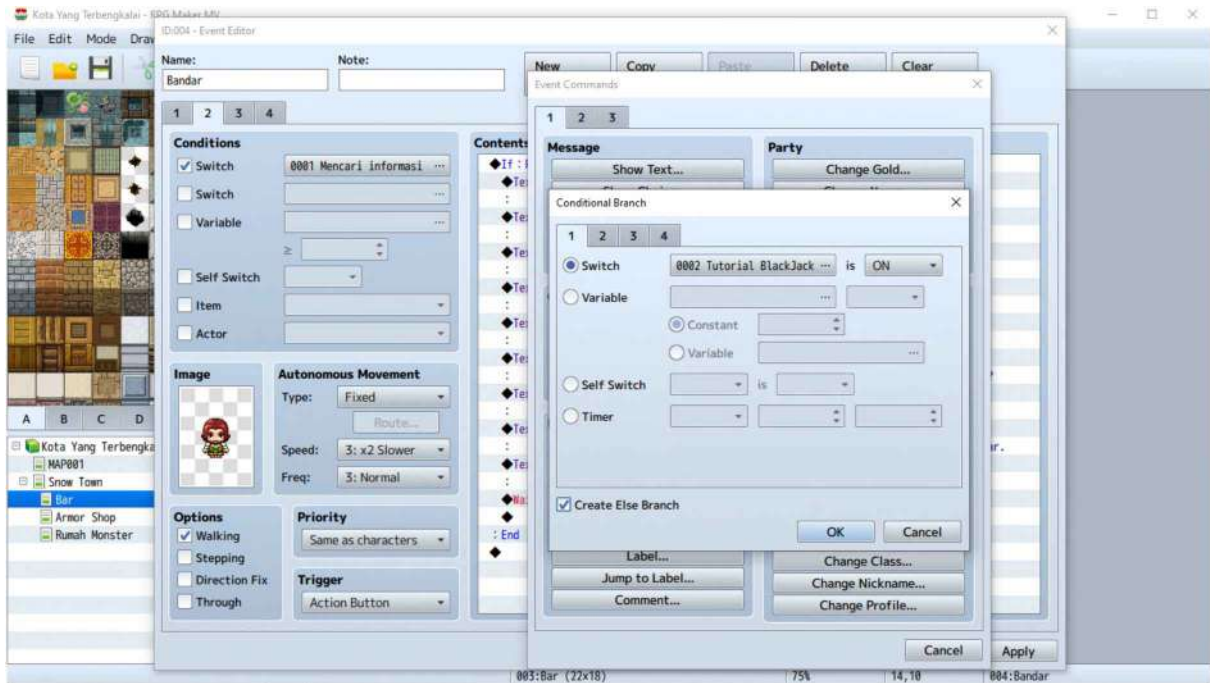
◆Text :Actor3(7), Window, Bottom

```

        :\\n<Izzi>Boleh berbagi informasi nggak pak?
♦Text :People3(4), Window, Bottom
        :\\n<Bandar>Informasi itu nggak gratis, memangnya kalian punya apa?
♦Text :Actor1(1), Window, Bottom
        :\\n<Kenta>Kami masih bocil, bapak berharap kami punya apa?
♦Text :People3(4), Window, Bottom
        :\\n<Bandar>Kalau kalian masih bocil, kenapa ingin tau mengenai monster itu?
♦Text :Actor2(6), Window, Bottom
        :\\n<Dicky>Karena menurut ramalan, kami akan mengalahkan monster itu.
♦Text :People3(4), Window, Bottom
        :\\n<Bandar>Hei nak kalau kalian ingin membuat lelucon, tulis dulu yang benar.
♦Text :Actor3(7), Window, Bottom
        :\\n<Izzi>KAMI TIDAK MELUCU, KAMI SERIUS.
♦Wait :60 frames
♦Text :People3(4), Window, Bottom
        :\\n<Bandar>Aku tidak peduli dan percaya dengan ramalan itu\\|. Tapi kalau kalian ingin
        :informasi itu\\.\\., kalian harus mengalahkanku dalam bermain Blackjack
        :sebagai gantinya.

```

Setelah itu masukkan *event Conditional Branch* untuk membuat variasi percakapan antara karakter pemain dengan Bandar seperti gambar di bawah ini.



Gambar 3.120 Membuat event *Conditional Branch* untuk variasi percakapan

Kemudian buat event di bawah baris *If: Tutorial BlackJack is ON* seperti tabel di bawah ini.

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>Oke, cukup mudah.

◆Text :People3(4), Window, Bottom

:\n<Bandar>Sekarang siapa yang ingin bermain?

◆Text :Actor3(7), Window, Bottom

:\n<Izzi>\n<Izzi>Biar pemain kami yang memutuskan\|. Caranya adalah pilih \C[6]FORMATION\C[0] di pause menu\.,

Setelah itu pilih salah satu dari kami dan letakkan di urutan teratas untuk

bermain.

◆Control Self Switch: A = ON

Untuk variasi percakapan kedua, buat event di bawah baris *Else* seperti tabel di bawah ini.

◆Text :Actor3(7), Window, Bottom

·\n<lzzi>GIMANA CARA MAINNYA?

◆Text :People3(4), Window, Bottom

·\n<Bandar>Cukup mudah\|. Kalian hanya perlu mempunyai kartu yang jumlahnya 21\.\.,
·lebih dari itu kalian kalah.

◆Text :Actor2(6), Window, Bottom

·\n<Dicky>Oke, cukup mudah.

◆Text :People3(4), Window, Bottom

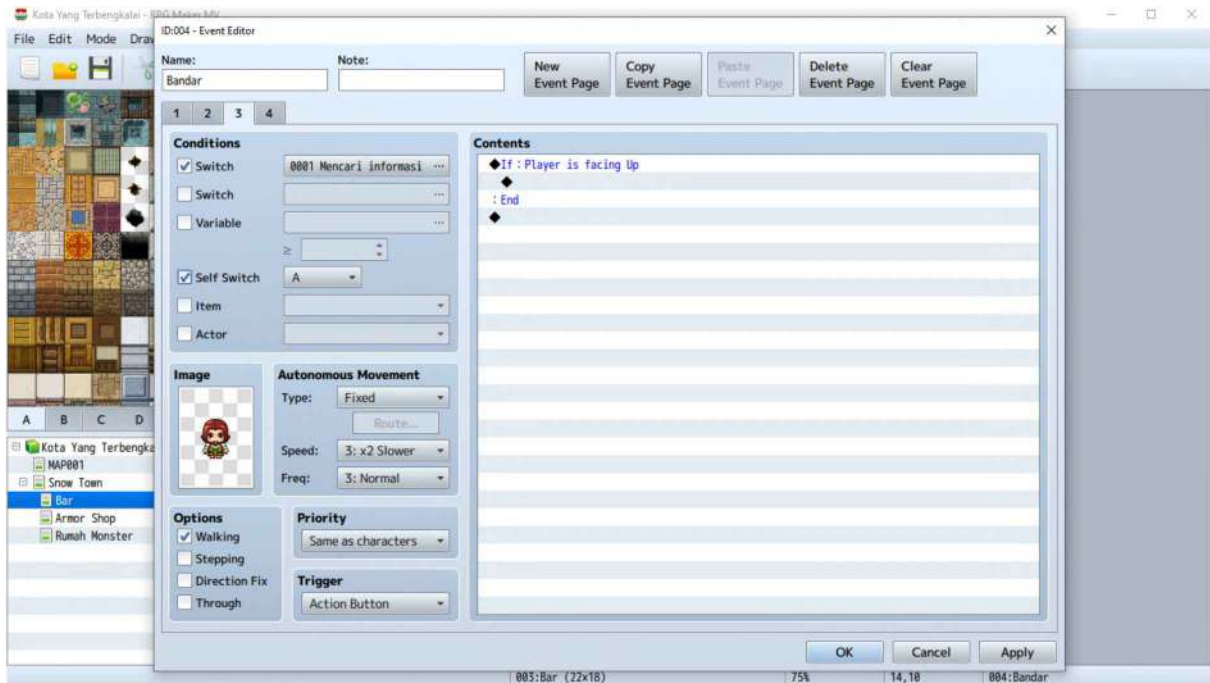
·\n<Bandar>Sekarang siapa yang ingin bermain?

◆Text :Actor3(7), Window, Bottom

·\n<lzzi>\n<lzzi>Biar pemain kami yang memutuskan\|. Caranya adalah pilih \C[6]FORMATION\C[0]
·di pause menu\.\.,
·Setelah itu pilih salah satu dari kami dan letakkan di urutan teratas untuk
·bermain.

Control Self Switch A = ON

Selanjutnya klik halaman tiga lalu centang tulisan *Switch 0001* Mencari informasi. dan *Self Switch A* di kotak *Conditions* seperti gambar di bawah ini.



Gambar 3.121. Mengaktifkan *Switch* dan *Self Switch* *Condition* bersamaan

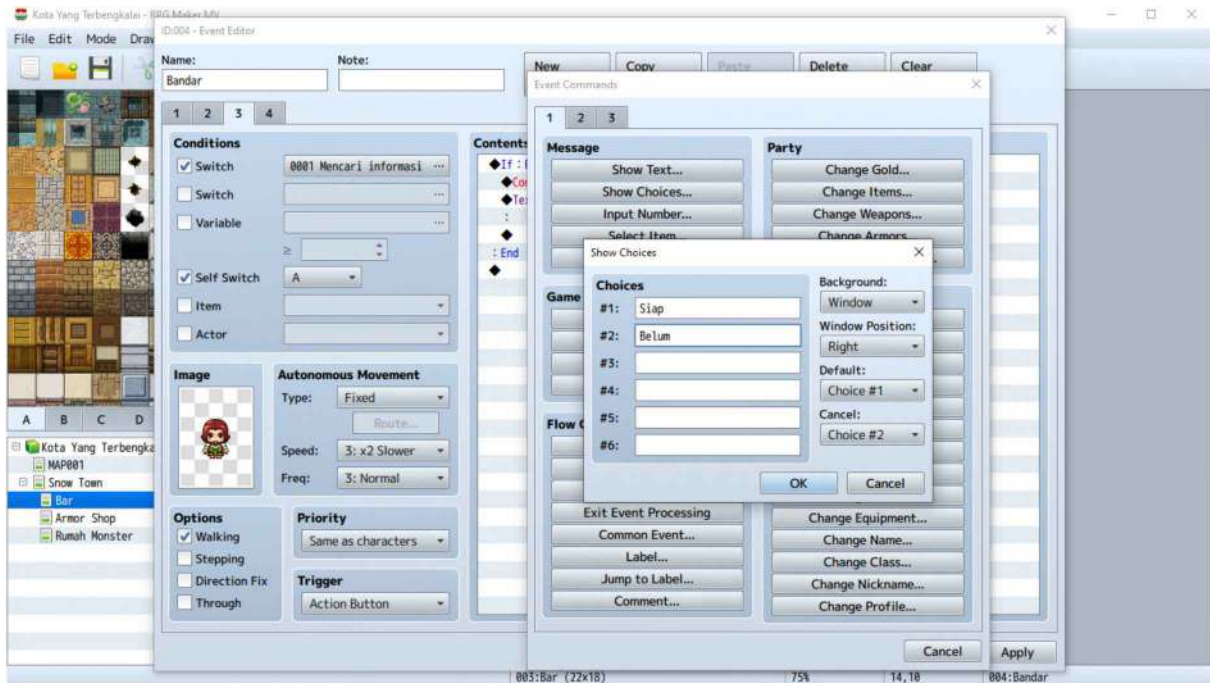
Setelah itu klik pada baris di bawah *If: Player is Facing Up* lalu buat *event* seperti gambar di bawah ini.

Control Variables: #0005 Kartu = 0

◆Text : People3(4), Window, Bottom

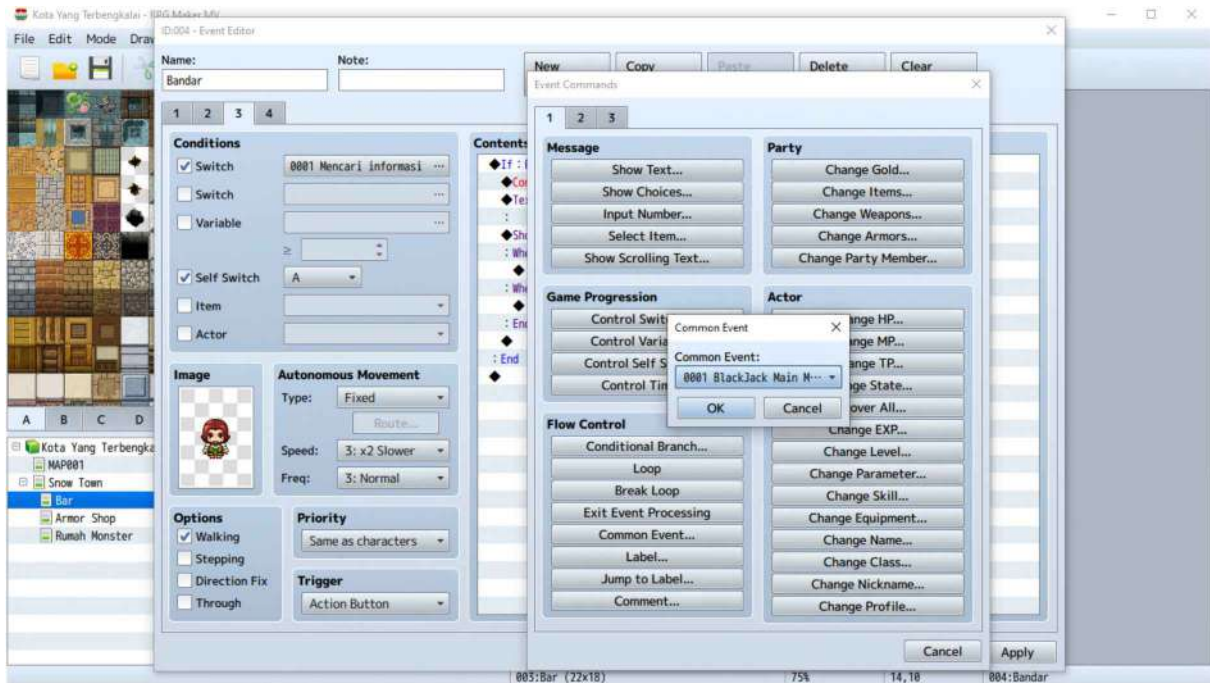
: \n<Bandar>Siap bermain?

Kemudian buat *event Show Choice* seperti ini.



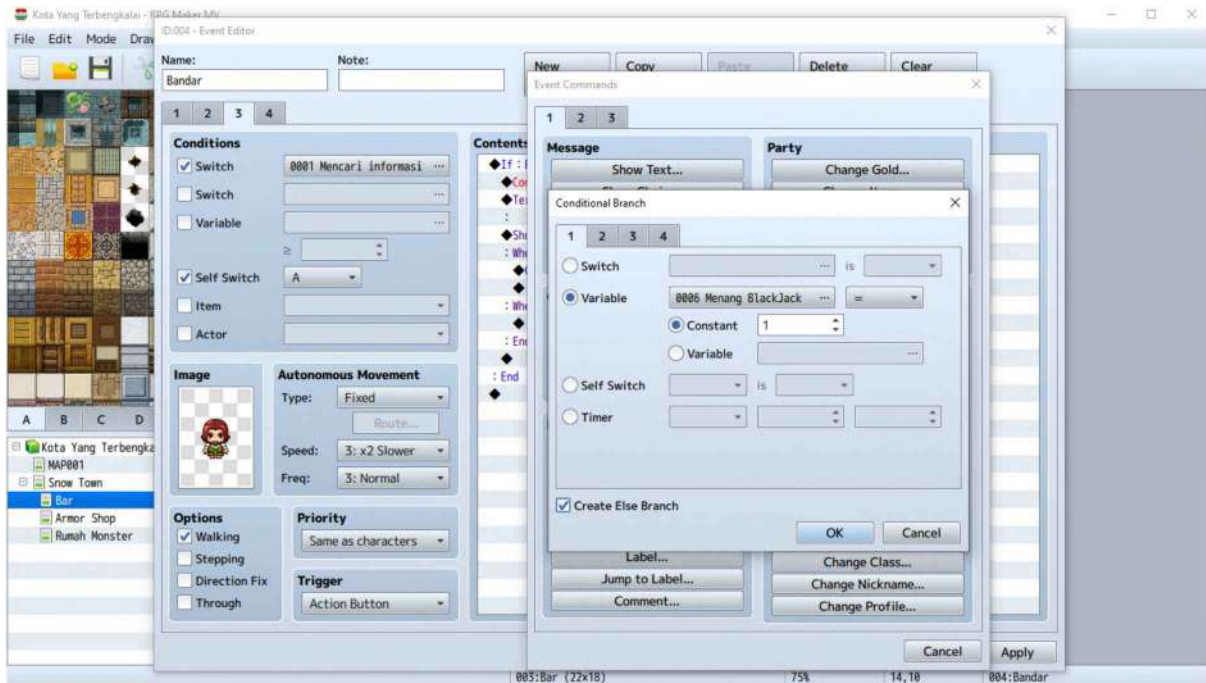
Gambar 3.122. Membuat *event Show Choice* setelah teks

Biarkan baris di bawah **When Belum** kosong lalu klik baris di bawah **When Siap** dan tambahkan *event Common Event* seperti di bawah ini.



Gambar 3.123. Membuat *Common Event BlackJack Main Mission*

Setelah itu masukkan *event Conditional Branch* untuk membuat *event* apabila pemain menang atau kalah bermain BlackJack.



Gambar 3.124. Membuat *Conditional Branch* apabila menang atau kalah bermain *BlackJack*

Kemudian buat event di bawah baris *If: Tutorial BlackJack is ON* seperti tabel di bawah ini.

◆Text :Actor3(7), Window, Bottom

:\n<lzizi>Yeay kita menang.

◆Text :Actor1(1), Window, Bottom

:\n<Kenta>Ayo pak bandar sesuai janjimu.

◆Text :People3(4), Window, Bottom

:\n<Bandar>Baik aku akan menempati janjiku\|. Apa yang ingin kalian ketahui?

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>Oke seperti apa monster itu?

◆Text :People3(4), Window, Bottom

:\n<Bandar>Namanya adalah Darklord\|. Wajahnya seperti manusia\.\., namun memiliki tanduk seperti

:Baphomet dan juga memiliki sayap.

◆Text :Actor1(1), Window, Bottom

:\\n<Kenta>Ngeri juga ya.

◆Text :Actor2(6), Window, Bottom

:\\n<Dicky>Gimana cara mengalahkannya?

◆Text :People3(4), Window, Bottom

:\\n<Bandar>Kalian salah kalau bertanya itu padaku\\|. Tapi aku kenal orang yang bisa menyiapkan
perlengkapan untuk kalian.

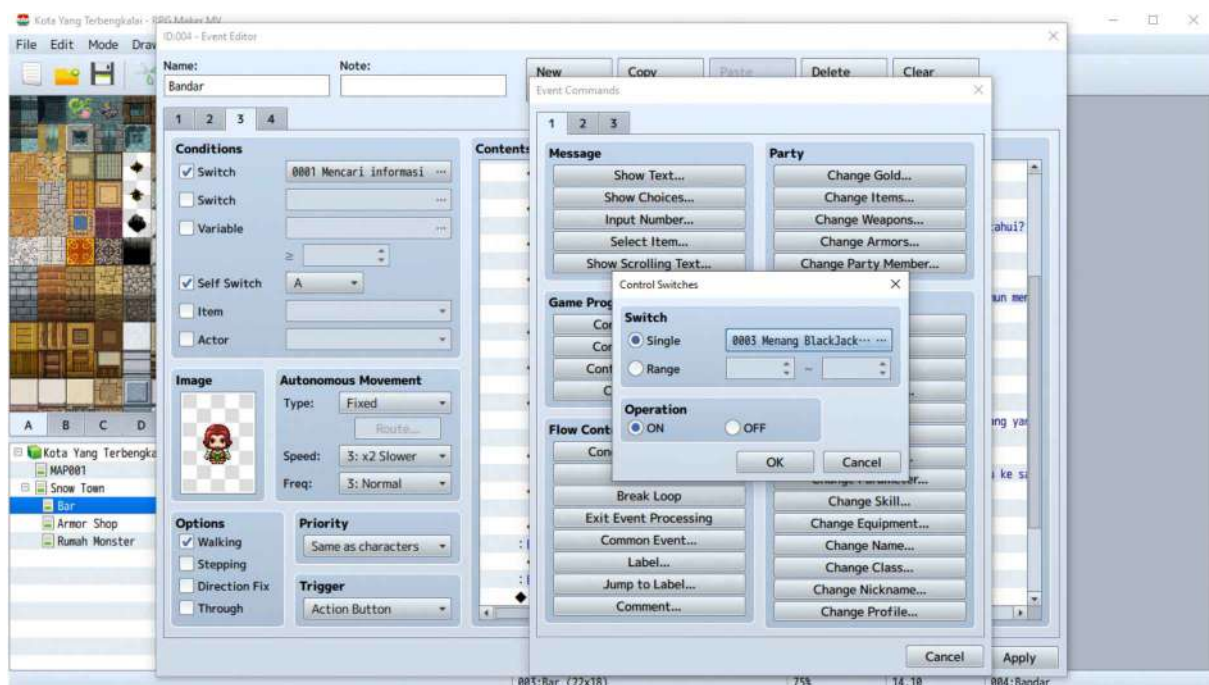
◆Text :People3(4), Window, Bottom

:\\n<Bandar>Ada toko di belakang bar ini. Setelah ini kalian bisa menuju ke sana.

◆Text :Actor3(7), Window, Bottom

:\\n<Izzi>Nice info, terima kasih pak bandar.

Setelah itu tambahkan *event Control Switch 0003 Menang BlackJack Main Mission* seperti gambar di bawah ini.



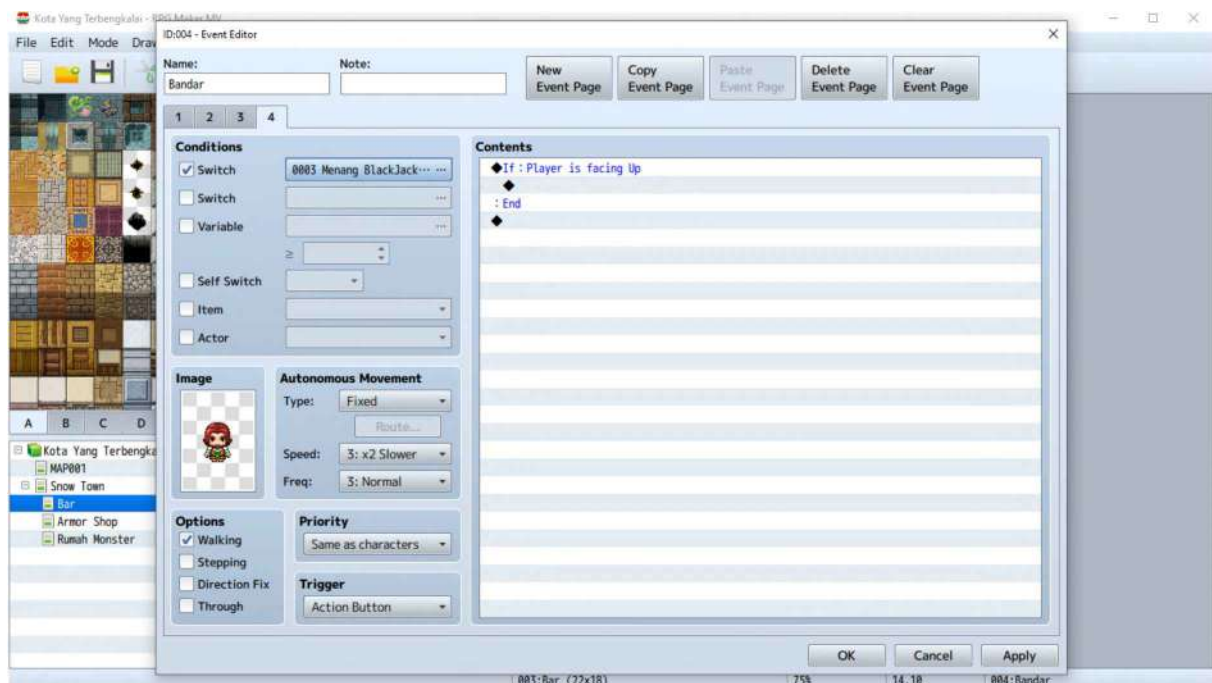
Gambar 3.125. Membuat *Control Switch 0003*

Kemudian tambahkan *event* di bawah baris *Else* seperti tabel di bawah ini.

◆Text :Actor1(1), Window, Bottom

: \n<Kenta>Ah gimana sih.

Selanjutnya klik halaman empat lalu centang tulisan *Switch* 0003 Menang *BlackJack Main Mission* di kotak *Conditions* seperti gambar di bawah ini.



Gambar 3.126. Mengaktifkan *Switch* 003 Menang *BlackJack Main Mission*

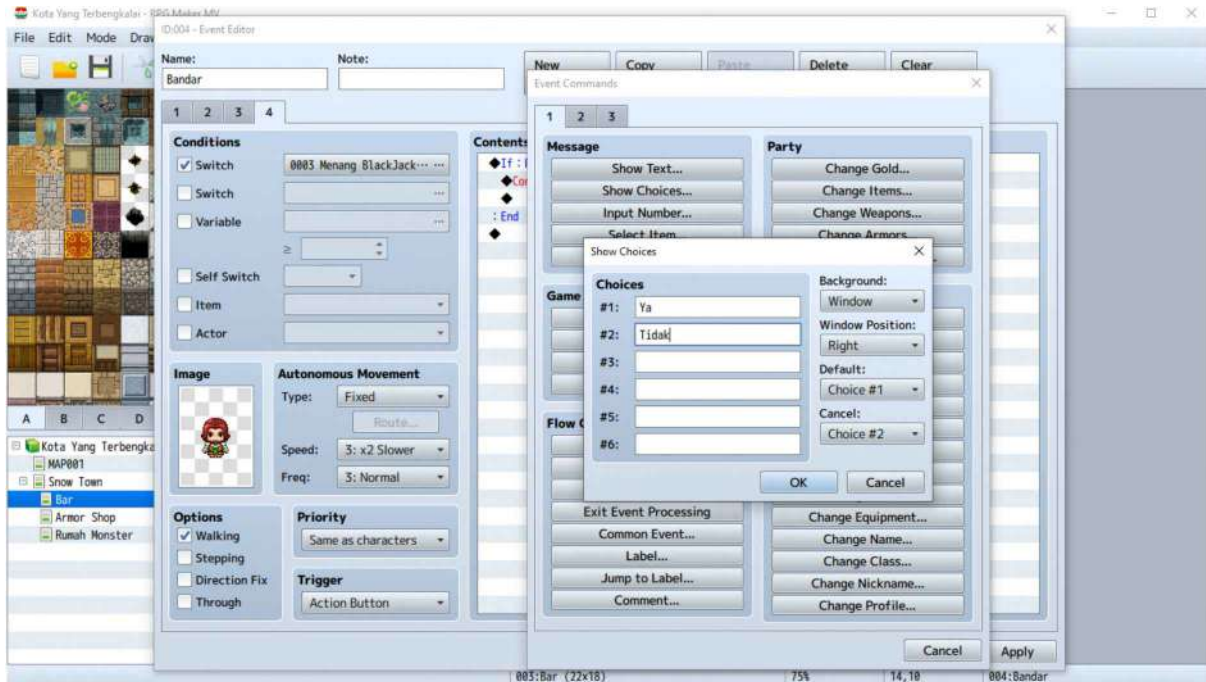
Setelah itu klik pada baris di bawah *If: Player is Facing Up* lalu buat *event* seperti gambar di bawah ini.

◆Control Variables: #0005 Kartu = 0

◆Text : People3(4), Window, Bottom

:\n<Bandar>Halo nak, mau main BlackJack lagi?

Kemudian *buat event Show Choice* seperti ini.



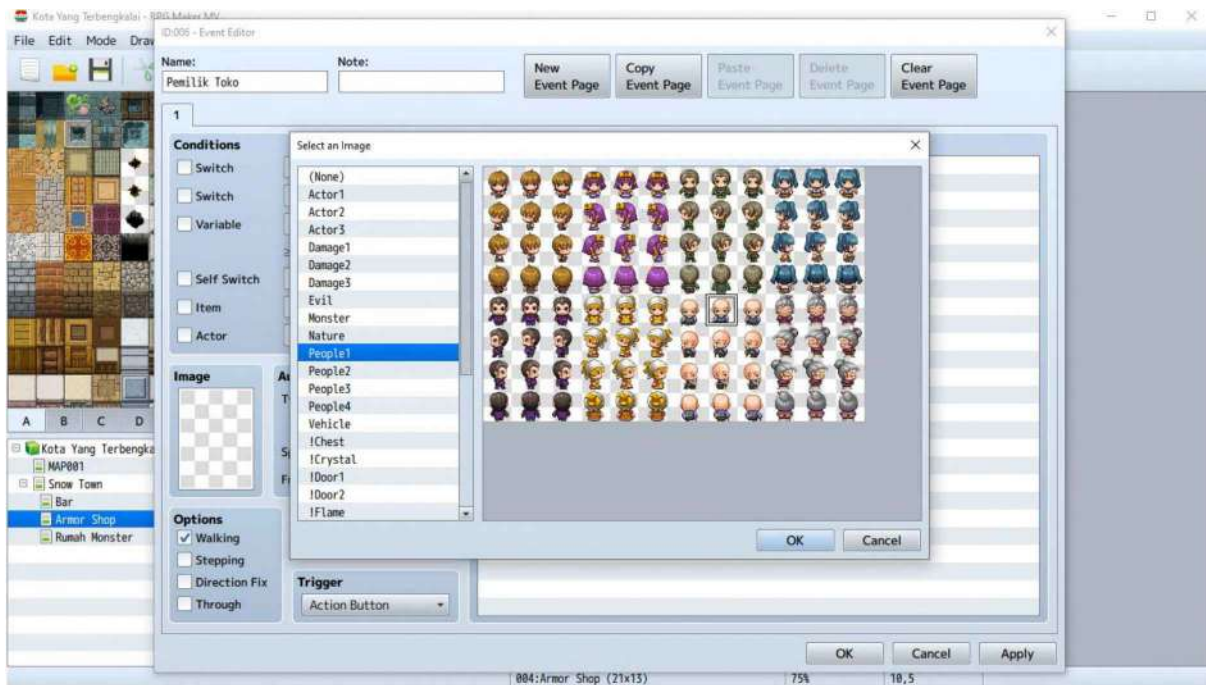
Gambar 3.127. Menambahkan *event Show Choice* untuk bermain BlackJack lagi

Biarkan baris di bawah **When Tidak** kosong lalu klik baris di bawah **When Ya** dan tambahkan *event Common Event BlackJack Side Mission*.

Pemilik Toko dan Dukun

Event selanjutnya yang dibuat adalah *event* di *map Armor Shop*. Klik *map Armor Shop* di sebelah kiri sampai *map* berganti tampilan menjadi *map Armor Shop*.

Setelah itu klik dua kali di koordinat (10,5) kemudian namai *event* ini **Pemilik Toko**. Klik dua kali di kolom *image* dan pilih *People1(6)* lalu klik *OK*.



Gambar 3.128. Membuat NPC Pemilik Toko

Kemudian klik *Copy Event Page* dan *Paste Event Page* dua kali untuk menduplikat halaman menjadi tiga. Setelah itu tambahkan *event* di kotak *Contents* halaman satu seperti tabel di bawah ini.

◆Text :People1(6), Window, Bottom

:\n<Pemilik Toko>Ada yang bisa kubantu?

◆Text :Actor1(1), Window, Bottom

:\n<Kenta>Kami ingin meminta perlengkapan untuk melawan monster\|.

:Kata bandar yang ada di bar, kami bisa memintanya ke sini.

◆Text :People1(6), Window, Bottom

:\\n<Pemilik Toko>Meminta? Hahaha lucu sekali\\.|. Tidak ada yang gratis, nak\\.|. Toko ini

:sudah sepi pengunjung dan kalian malah datang ke sini untuk mengemis.

◆Text :Actor2(6), Window, Bottom

:\\n<Dick>Makanya itu cuma kami satu-satunya yang mengunjungi toko bapak.

◆Text :Actor3(7), Window, Bottom

:\\n<Izzi>Betul\\.\\., kami ini orang yang ada di ramalan yang bisa mengalahkan monster itu.

◆Text :People1(6), Window, Bottom

:\\n<Pemilik Toko>Bocil seperti kalian bisa mengalahkan monster itu?\\.\\.. Aku tidak percaya.

◆Text :Actor3(7), Window, Bottom

:\\n<Izzi>Kami tidak butuh kepercayaan bapak, kami hanya butuh perlengkapan saja.

◆Text :People1(6), Window, Bottom

:\\n<Pemilik Toko>Apa buktinya kalau kalian adalah orang yang ada di ramalan itu?

◆Text :Actor1(1), Window, Bottom

:\\n<Kenta>Salah satu dari kami ada yang namanya Dick.

◆Wait :60 frames

◆Text :People1(6), Window, Bottom

:\\n<Pemilik Toko>Apakah mengarang kalian cuma segitu saja?

Fadein Screen

Control Switches: #0005 Dukun datang = ON

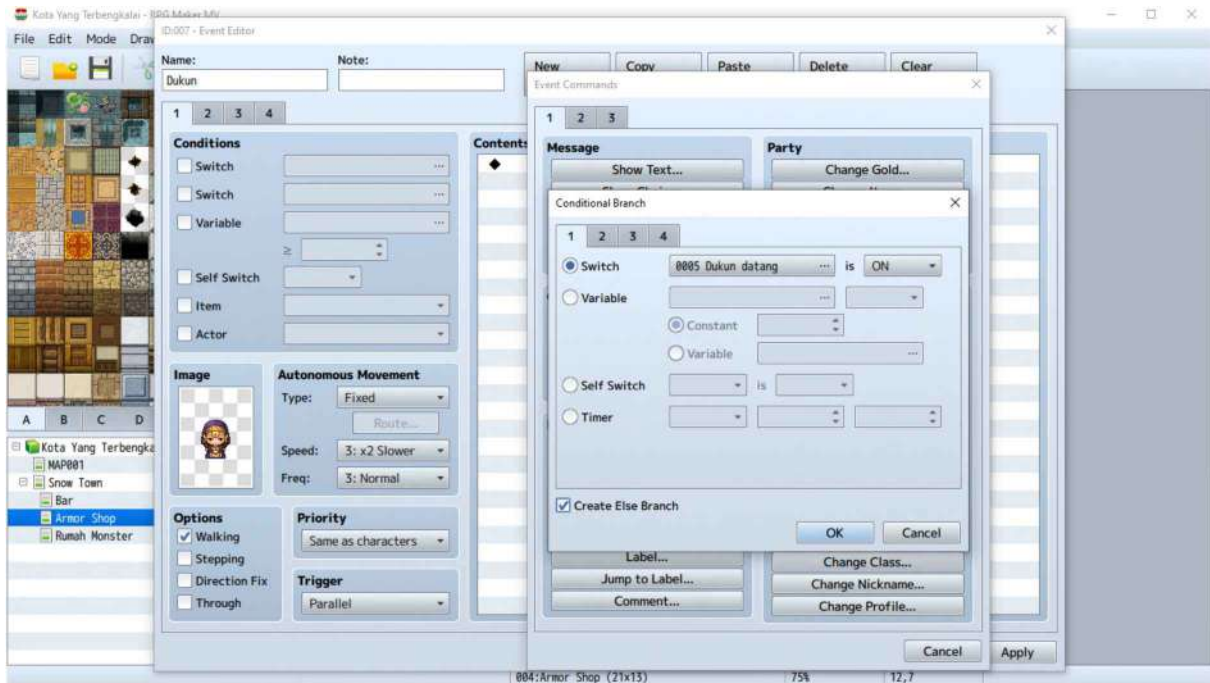
◆Wait :120 frames

Tutup jendela *Event Editor* Pemilik Toko terlebih dahulu dengan klik *Apply* lalu *OK*. Kemudian klik dua kali di koordinat (12,7) dan beri nama *event* ini **Dukun**. Klik dua kali di kolom *image* dan pilih *People4(6)* lalu klik *OK*.



Gambar 3.129. Membuat NPC Dukun

Ubah *Trigger*-nya menjadi *Parallel*. Kemudian klik *Copy Event Page* dan *Paste Event Page* tiga kali untuk menduplikat halaman menjadi empat. Setelah itu tambahkan *event Conditional Branch* di kotak *Contents* halaman satu seperti gambar di bawah ini.



Gambar 3.130. Membuat *event Conditional Branch* Dukun

Selanjutnya tambahkan *event* di bawah baris *If*: Dukun Datang is ON.

◆Set Movement Route : Dukun (**Wait**)
 : ◆Transparent OFF
 Control Self Switches: A = ON

Kemudian tambahkan *event* di bawah baris *Else*.

◆Set Movement Route : Dukun (**Wait**)
 : ◆Transparent ON

Setelah itu tutup jendela *Event Editor* Dukun lalu buka kembali *Event Editor* Pemilik Toko kemudian buat *event* di bawah baris *Wait 120 frames* seperti di bawah ini.

◆Set Movement Route :Dukun (**Wait**)
:◇Turn Left
:◇Transparent OFF

◆Play SE :Teleport (25, 100, 0)

◆Wait :180 frames

Fadein Screen

◆Wait :60 frames

◆Text :**People4(7), Window, Bottom**

:\\n<Dukun>Hei botak\\.\\. , berikan saja perlengkapannya\\|. Bocil-bocil ini memang orang
:yang ada di ramalanku.

◆Text :**Actor3(7), Window, Bottom**

:\\n<Izzi>Loh, sejak kapan dia ada di sini?

◆Text :**Actor1(1), Window, Bottom**

:\\n<Kenta>Memang dukun sakti.

◆Text :**People1(6), Window, Bottom**

:\\n<Pemilik Toko>Terserahmu saja\\.\\. , kalau memang ramalanmu seperti itu.

◆Text :**Actor2(6), Window, Bottom**

:\\n<Dicky>Nah gitu dong.

◆Text :**Actor1(1), Window, Bottom**

:\\n<Kenta>Apa aja nih yang harus kami ambil?

◆Text :**People1(6), Window, Bottom**

:\\n<Pemilik Toko>Hmmm\\.\\.\\.\\. apa ya?\\| Kalian bisa ambil pedang, panah, liontin, dan perisai
:di meja kiri kalian.

◆Text :**Actor3(7), Window, Bottom**

:\\n<Izzi>Liontin? \\.\\. Yang bener aja, pak\\|. Kita mau ngelawan monster lho.

◆Text :**People1(6), Window, Bottom**

:\\n<Pemilik Toko>Sabar\\|. Liontin itu nanti bisa menjadi tongkat sihir yang bisa membantu kalian.

◆Text :Actor3(7), Window, Bottom

·\n<Izzi>Wah keren.

◆Text :Actor2(6), Window, Bottom

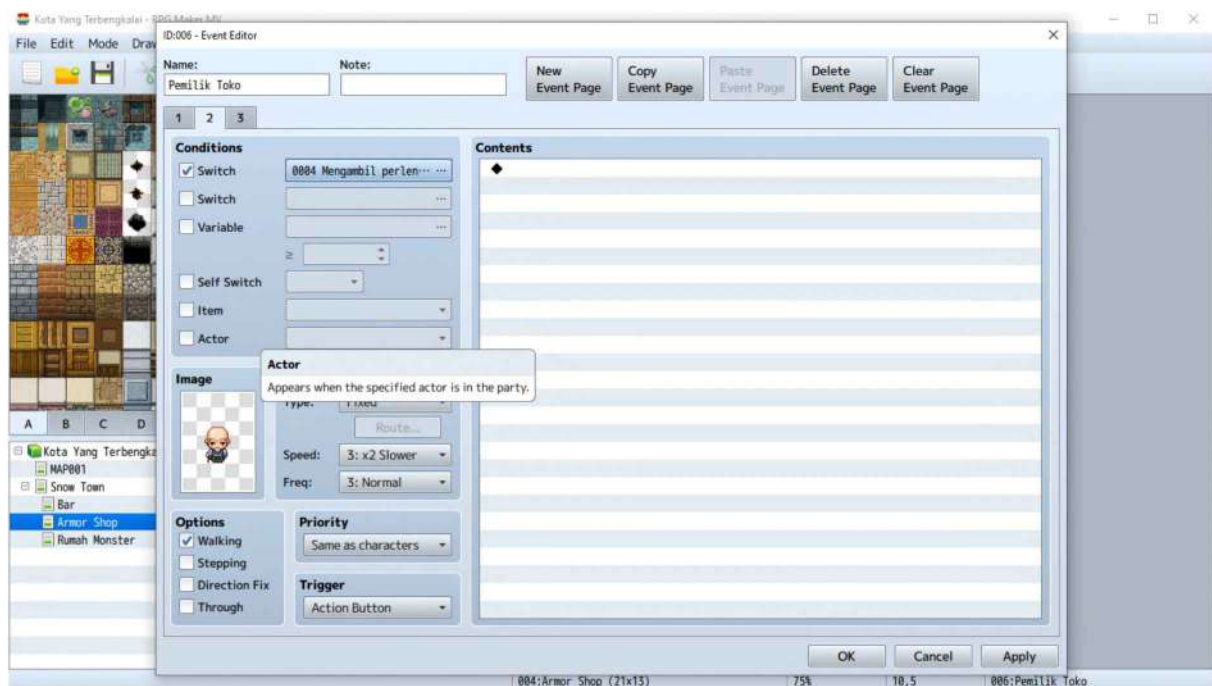
·\n<Dicky>Ayo kita ambil dulu.

◆Set Movement Route :Dukun (Wait)

·◆Turn Down

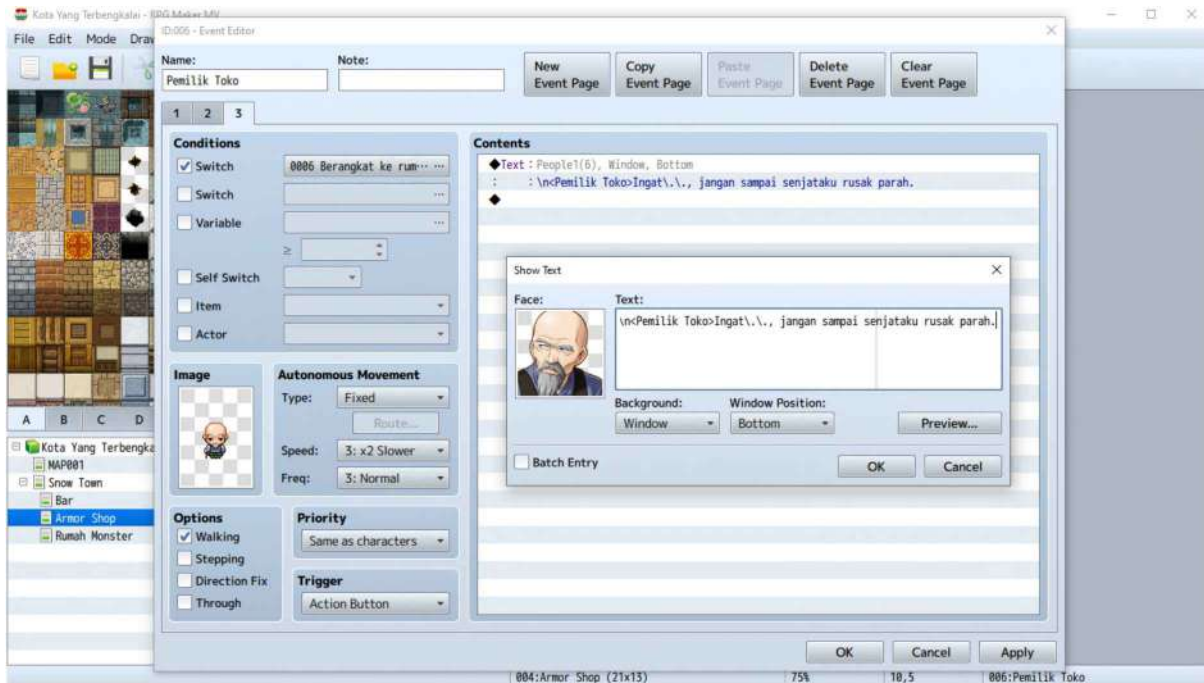
Control Switches: #004 Mengambil perlengkapan = ON

Setelah itu klik pada halaman dua dan atur menjadi seperti ini.



Gambar 3.131. Membuat *Control Switch 0004*

Kemudian atur halaman tiga menjadi seperti ini.



Gambar 3.132. Halaman tiga *Event Editor* Pemilik Toko

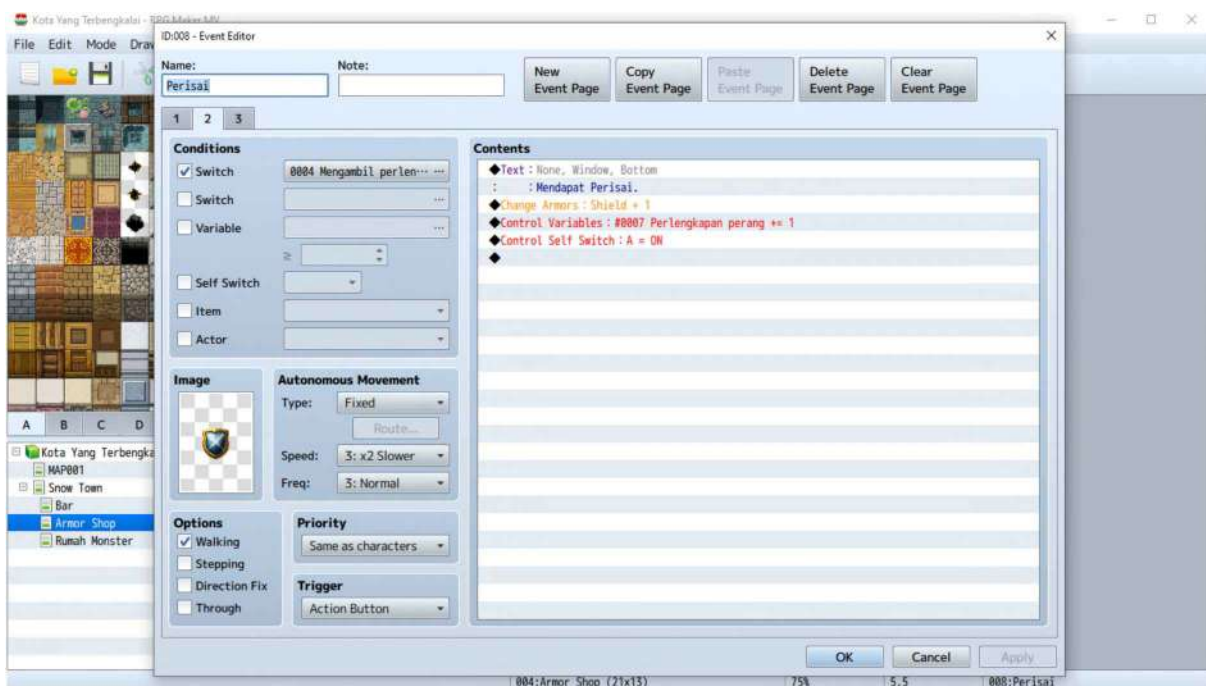
Mengambil Objek

Event yang akan dibuat selanjutnya adalah *event* mengambil objek. Klik dua kali pada koordinat (5,5) lalu beri nama *event* ini **Perisai** dan klik dua kali di kolom *image* kemudian pilih gambar perisai yang ada *Tilesets C* seperti gambar di bawah ini lalu klik *OK*.



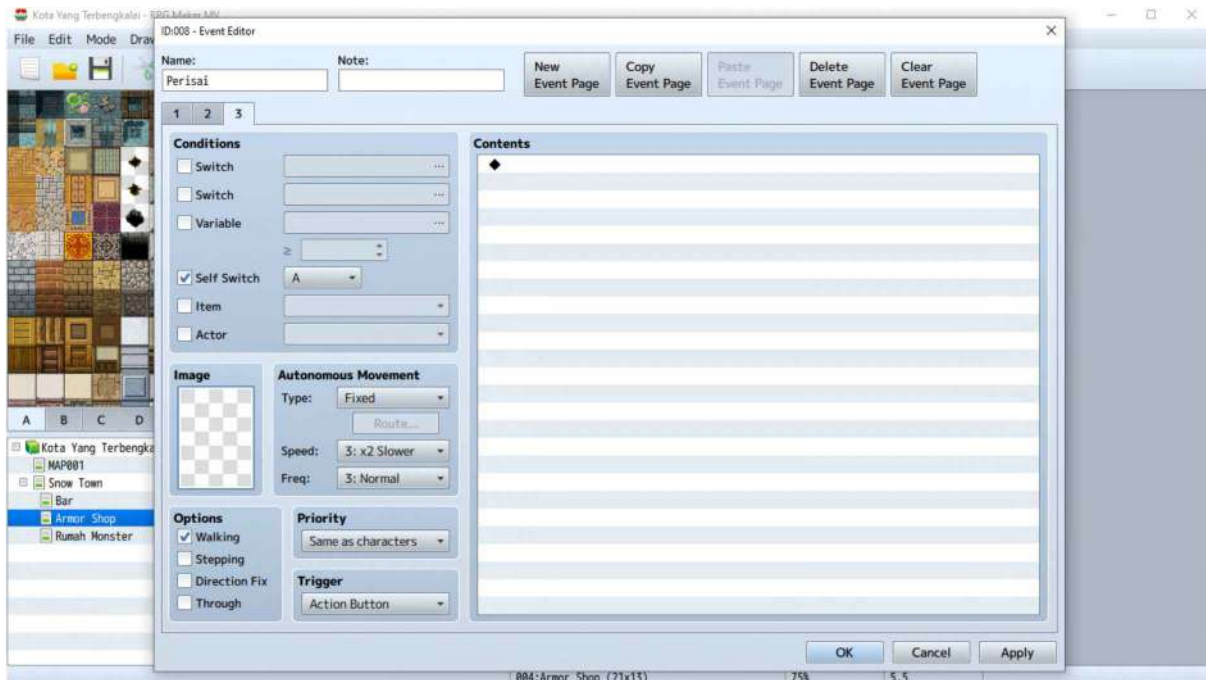
Gambar 3.133. Membuat *event* mengambil objek perisai

Ubah *Priority*-nya menjadi *Same as Characters*. Kemudian klik *Copy Event Page* dan *Paste Event Page* dua kali untuk menduplikat halaman menjadi tiga. Setelah itu klik halaman dua dan atur seperti gambar di bawah ini.



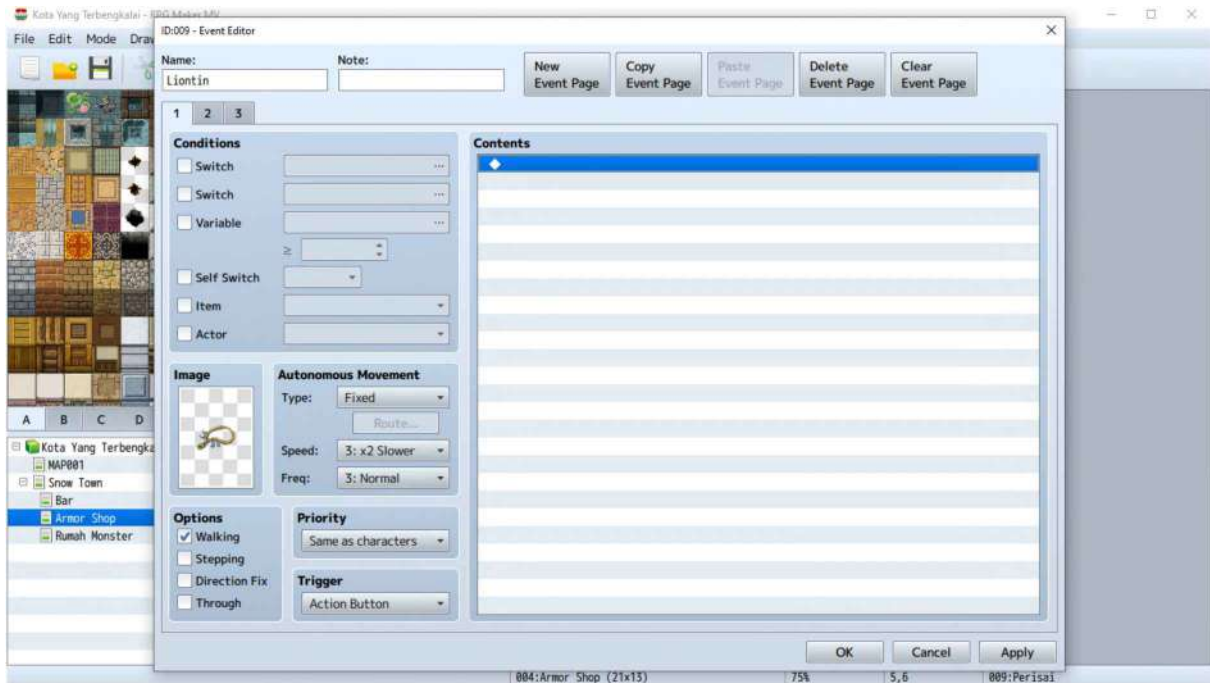
Gambar 3.134. Halaman dua *event* mengambil objek perisai

Selanjutnya klik halaman tiga lalu atur menjadi seperti ini.



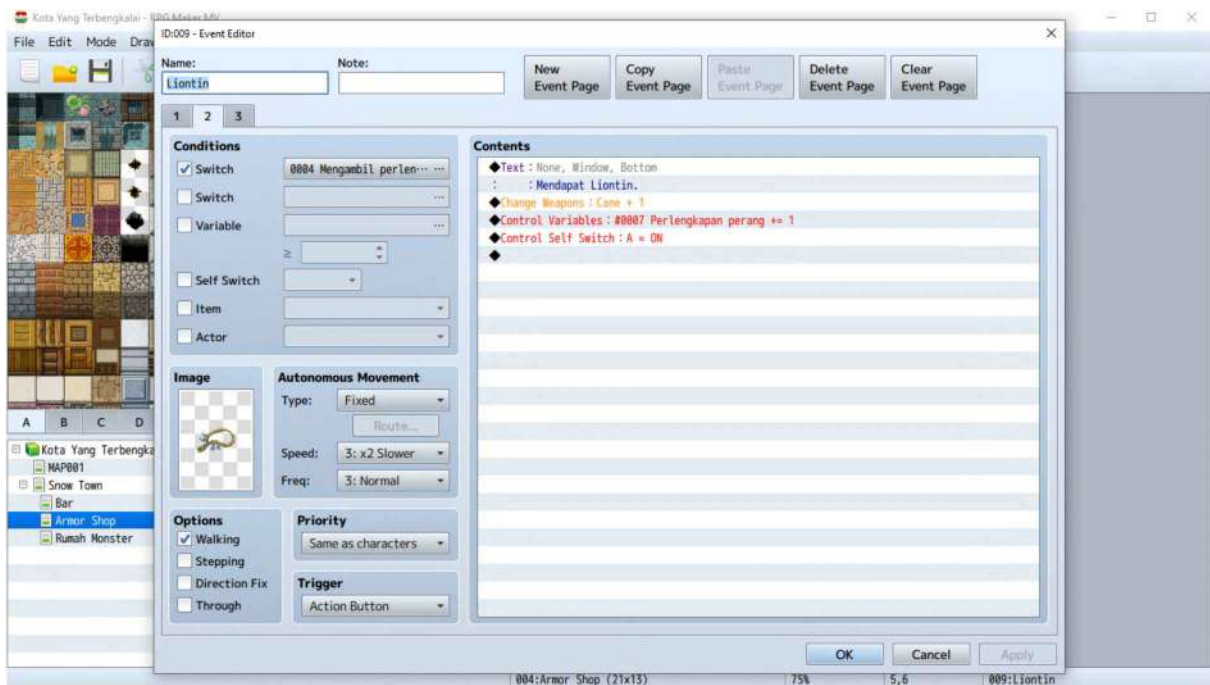
Gambar 3.135. Halaman tiga *event* mengambil objek perisai

Langkah selanjutnya adalah meng-*copy* objek perisai dengan cara klik kanan → *Copy* atau menekan Ctrl+C kemudian klik kanan pada koordinat (5,6) lalu *paste*. Setelah itu klik dua kali untuk objek di koordinat (5,6) lalu ubah nama *event*-nya menjadi **Liontin**. Klik kolom *image* kemudian pilih gambar *pendant* yang ada *Tilessets C* seperti gambar di bawah ini lalu klik *OK*.



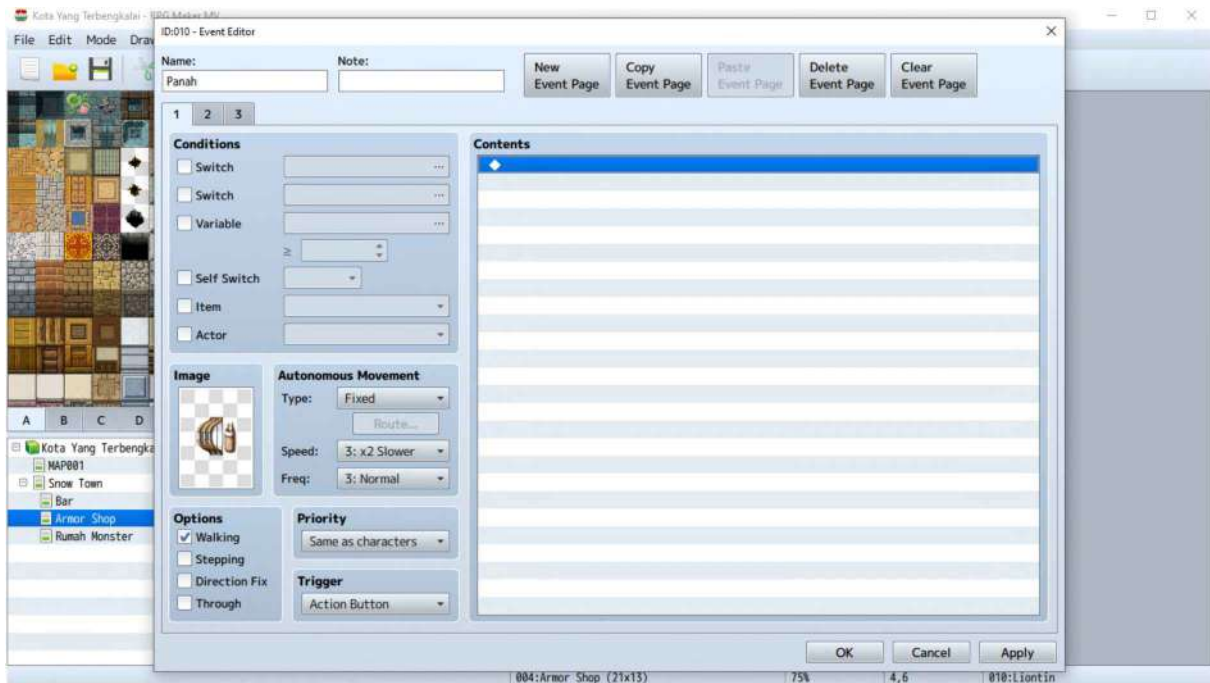
Gambar 3.136. Halaman satu *event* mengambil objek liontin

Setelah itu klik halaman dua dan atur seperti gambar di bawah ini.

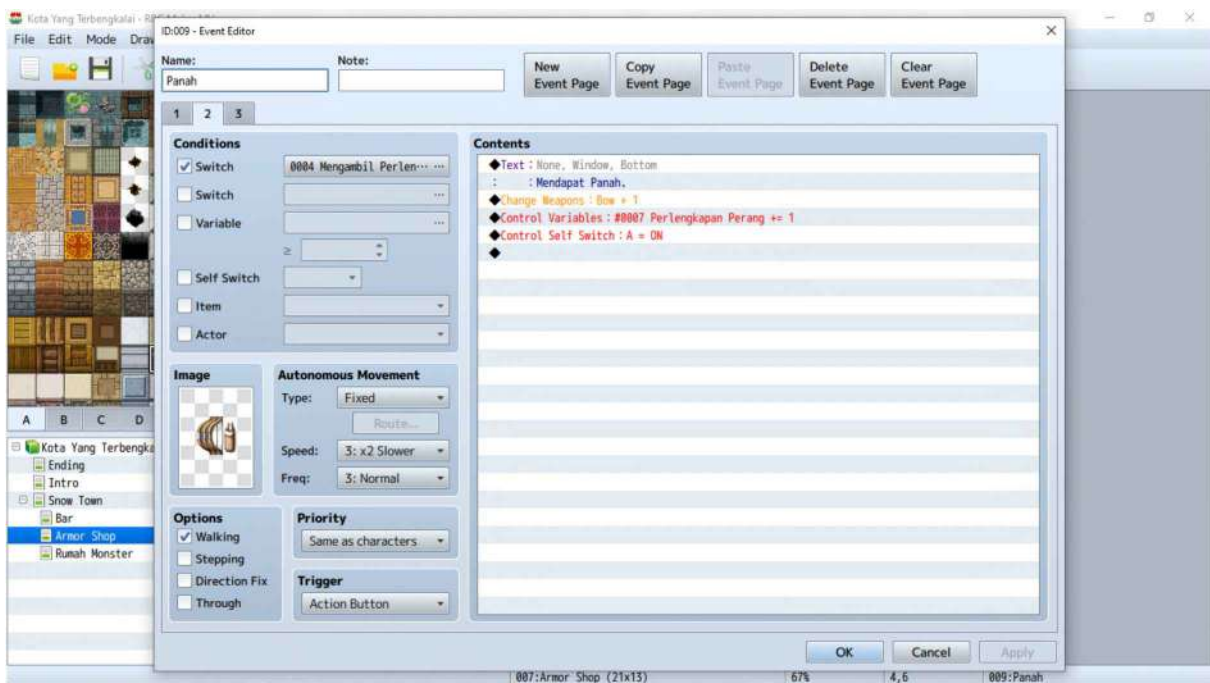


Gambar 3.137. Halaman dua *event* mengambil objek liontin

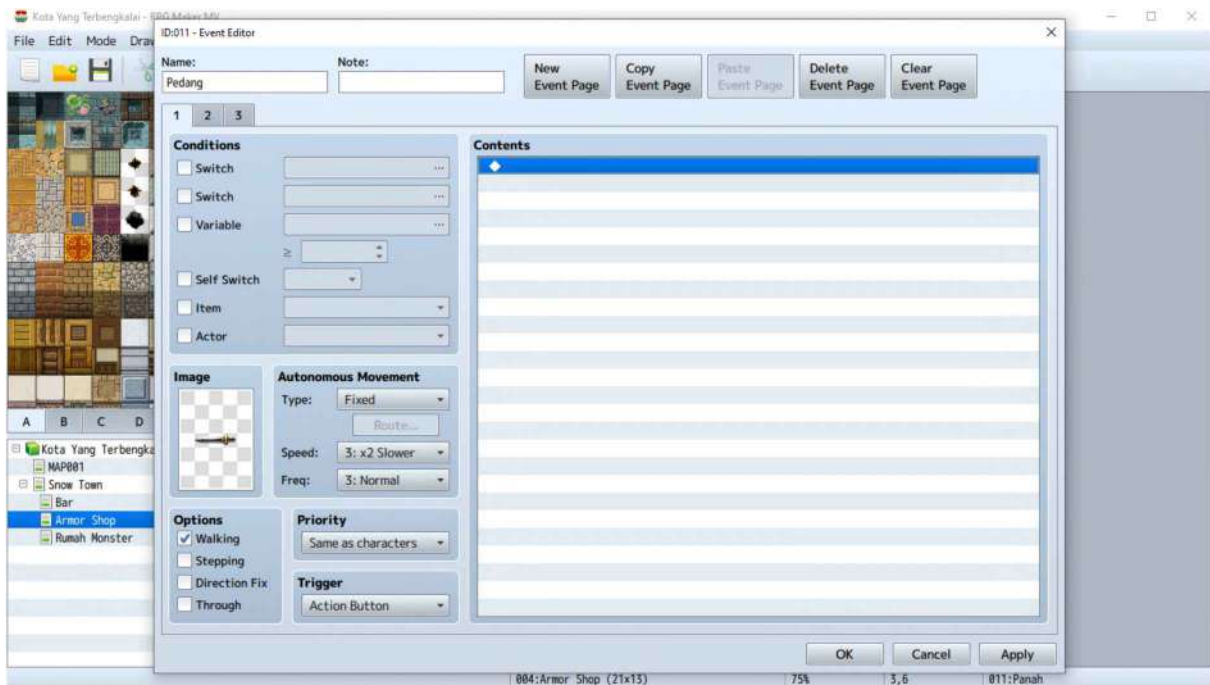
Lakukan cara di atas untuk objek pedang dan panah.



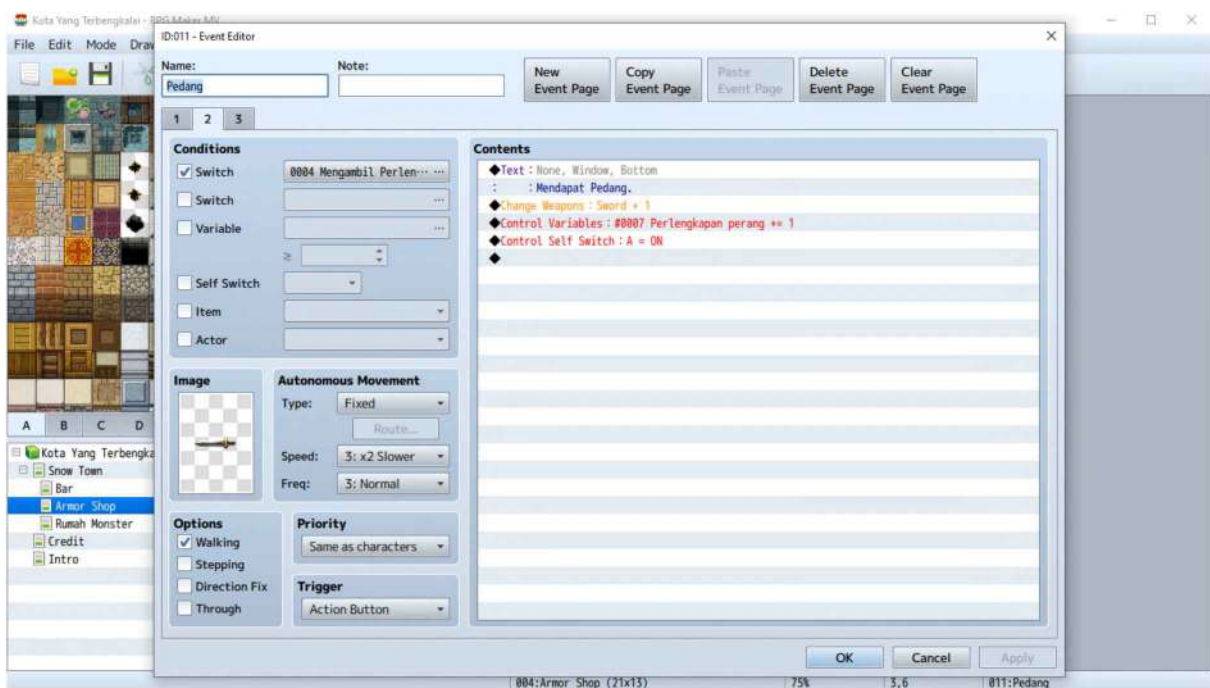
Gambar 3.138. Halaman satu *event* mengambil objek panah



Gambar 3.139. Halaman dua *event* mengambil objek panah



Gambar 3.140. Halaman satu *event* mengambil objek pedang

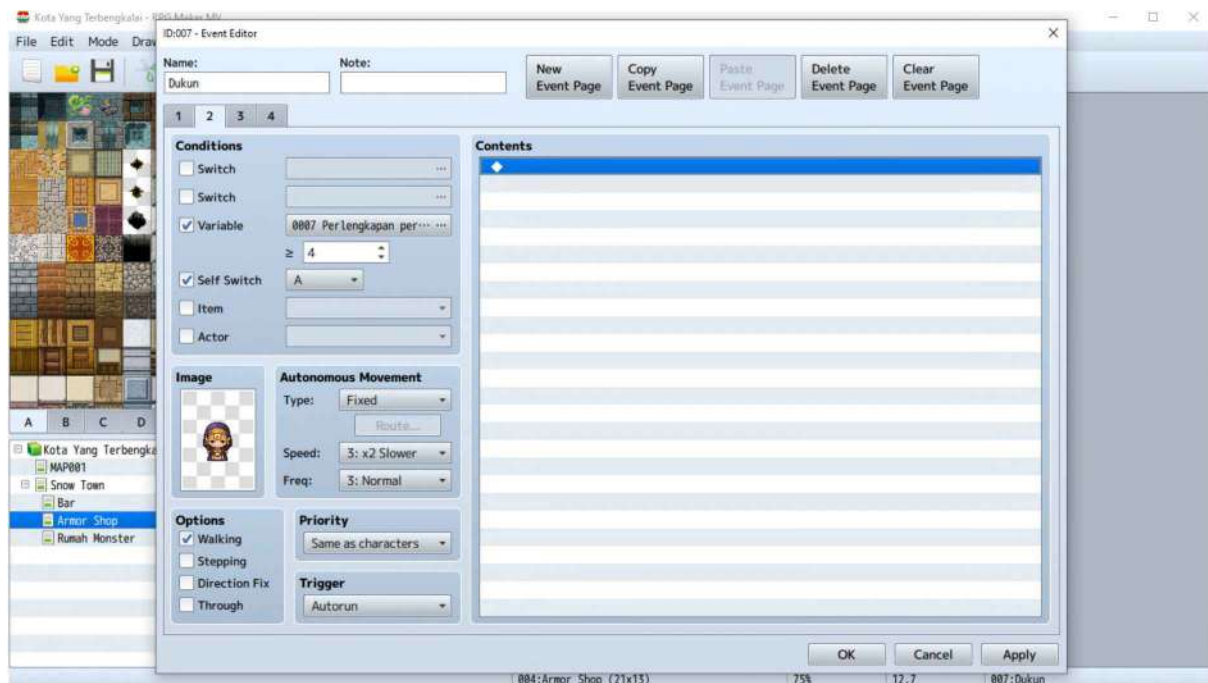


Gambar 3.141. Halaman dua *event* mengambil objek pedang

Menambah Skill

Setelah selesai mengambil objek, *NPC* Dukun akan memberikan masing-masing karakter *skill* untuk persiapan

melawan *monster*. Klik dua kali pada *NPC Dukun* kemudian klik halaman dua lalu centang *Variable 0007* Perlengkapan perang dan isi ≥ 4 . Selanjutnya centang *Self Switch A* dan ubah *Trigger*-nya menjadi *Autorun*.



Gambar 3.142. Halaman dua *event NPC Dukun*

Kemudian tambahkan *event* seperti di bawah ini.

◆Wait :60 frames

◆Text :People4(7), Window, Bottom

: \n<Dukun>Karena kalian sudah mendapatkan perlengkapan\|. Kemarilah\.\, aku akan memberikan

· kalian kekuatan tambahan.

Control Self Switch A = ON

Selanjutnya klik halaman tiga lalu centang pada *Control Self Switch B* dan ubah *Trigger*-nya menjadi *Action Button*. Kemudian tambahkan *event* seperti di bawah ini.

Show Balloon Icon: Dukun, Cobweb (Wait)

◆Text : People4(7), Window, Bottom

: \n<Dukun>Specialis Revelio.

◆Play SE : Powerup (35, 100, 0)

Show Balloon Icon: Player, Exclamation (Wait)

Change Skill: Kenta, + Triple Attack

Change Skill: Izzi, + Spark

Change Skill: Izzi, + Fire

Change Skill: Dicky, + Heal

Change Skill: Dicky, + Double Attack

◆Text : None, Window, Bottom

: Kenta mendapat skill Triple Attack

◆Text : None, Window, Bottom

: Izzi mendapat skill Spark dan Fire

◆Text : None, Window, Bottom

: Dicky mendapat skill Heal dan Double Attack

◆Text : People4(7), Window, Bottom

: \n<Dukun>Sekarang kalian sudah siap untuk menghadapi monster itu\|. Semoga ramalan itu terwujud.

◆Text : People1(6), Window, Bottom

: \n<Pemilik Toko>Perlengkapan kalian tidak bisa digunakan otomatis\|., jadi kalian harus memakainya

: sendiri\|. Pergi ke menu \C[6]EQUIP\C[0] yang ada di pause menu\|., kemudian gunakan untuk kalian masing-masing.

◆Text : People1(6), Window, Bottom

:\n<Pemilik Toko>Aku tidak peduli kalau kalian mati, tapi jangan sampai merusak senjataku
terlalu parah.

◆Text : Actor3(7), Window, Bottom

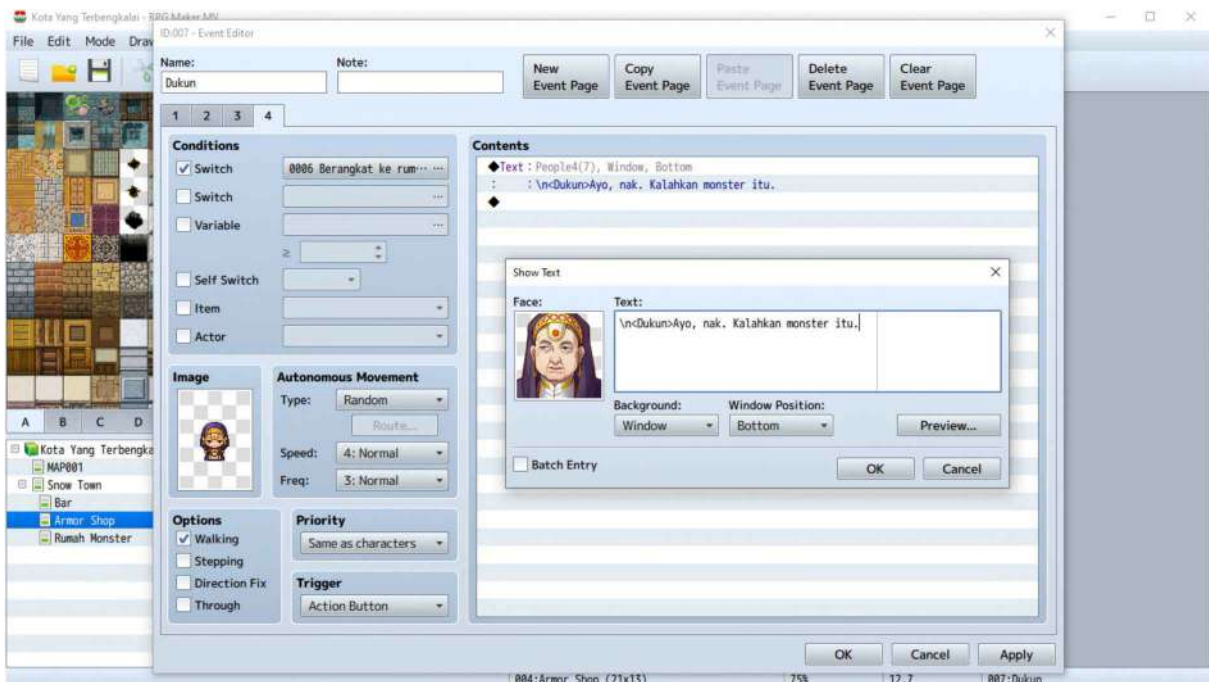
:\n<Izzi>Siap kakek, semoga harimu menyenangkan.

◆Text : Actor2(6), Window, Bottom

:\n<Dicky>Oke semua sudah siap, saatnya berangkat.

Control Switches: #0006 Berangkat ke rumah monster = ON

Untuk halaman empat Dukun buat seperti di bawah ini.



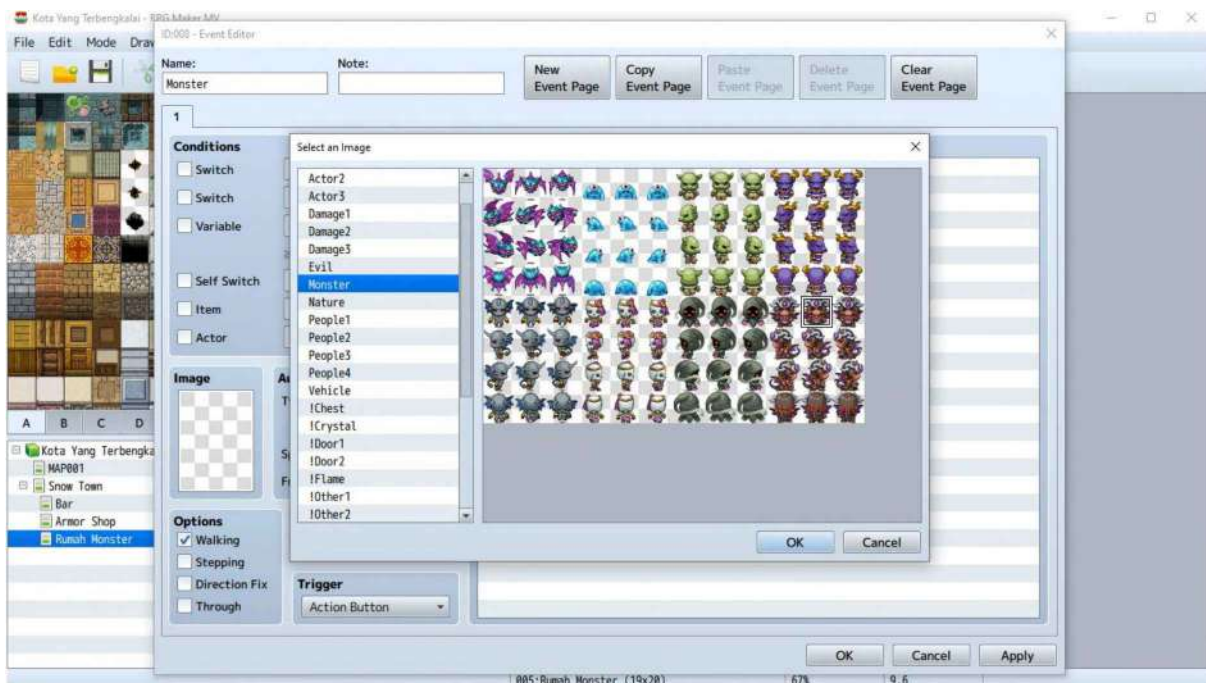
Gambar 3.143. Halaman empat *event* NPC Dukun

Rumah Monster

Setelah selesai membuat *event* pintu Rumah *Monster*, langkah selanjutnya adalah membuat *event* di dalam Rumah *Monster*.

Klik *map* Rumah *Monster* di sebelah kiri sampai *map* berganti tampilan menjadi *map* Rumah *Monster*.

Setelah itu klik dua kali pada koordinat (9,6) lalu beri nama *eventnya* dengan nama **Monster**. Klik *Copy Event Page* dan *Paste Event Page* sehingga terdapat dua halaman *Event Editor* lalu klik kolom *Image* dan pilih seperti gambar di bawah ini.



Gambar 3.144. Menambahkan *NPC Monster*

Kemudian klik halaman dua dan centang *Switch 0007 Monster* kalah lalu klik *OK*.

Selanjutnya klik dua kali di tempat yang tidak bisa disentuh pemain. Sebagai contoh saya menggunakan koordinat (12,19).

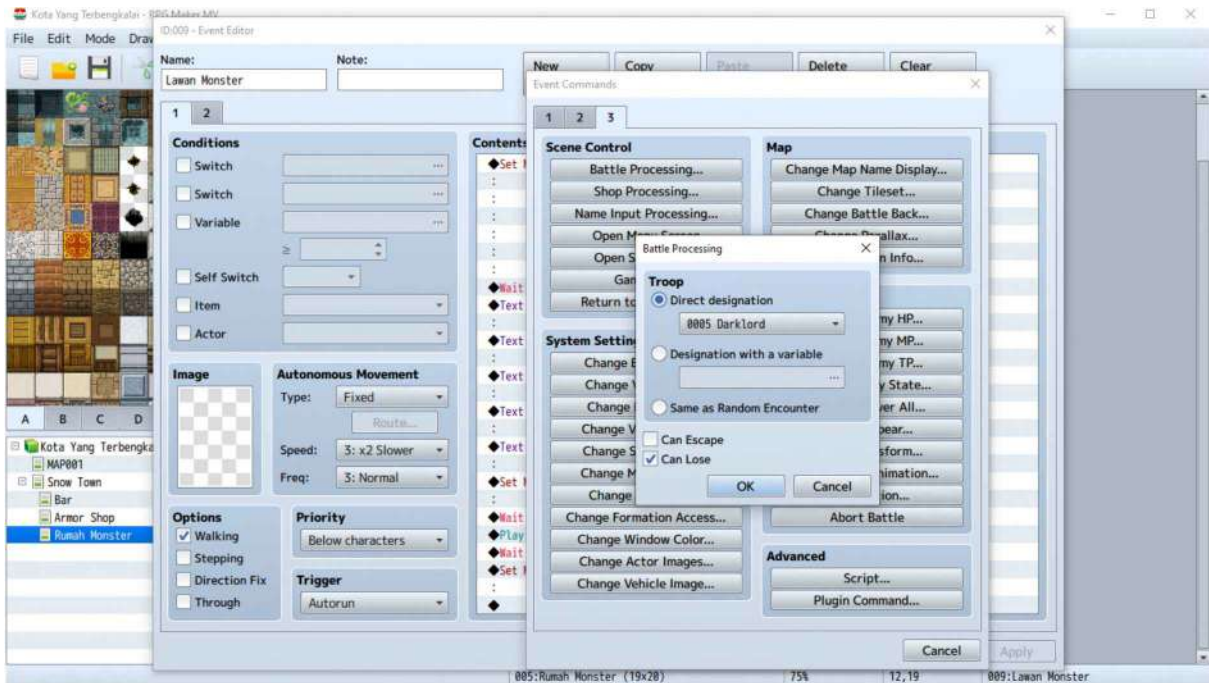
Setelah jendela *Event Editor* muncul, beri nama *eventnya* dengan nama **Lawan Monster**. Kemudian klik *Copy Event Page* lalu *Paste Event Page* sehingga terdapat dua halaman *Event Editor*. Klik halaman satu dan atur *Triggernya* menjadi *Autorun* lalu buat *event* seperti tabel di bawah ini.

◆Set Movement Route	: Player (Wait)
	: ◇ Move Up
	◇ Move Up
	◇ Move Up
	◇ Move Up
	◇ Move Up
	◇ Move Up
	◇ Move Up
◆Wait	: 120 frames
◆Text	: Monster(7), Window, Bottom
	: \n<Darklord>Jadi kalian\.\., orang yang dimaksud ramalan itu.
◆Text	: Actor3(7), Window, Bottom
	: \n<lizzi>Betul\.\.. Kami akan mengalahkanmu\ , makhluk jelek.
◆Text	: Monster(7), Window, Bottom
	: \n<Darklord>HAHAHA bermimpilah sebelum kalian mati.
◆Text	: Actor2(6), Window, Bottom
	: \n<Dicky>Kita lihat siapa yang mati.
◆Text	: Monster(7), Window, Bottom
	: \n<Darklord>Bocil ingusan seperti kalian memang membuatku terhibur.
◆Set Movement Route	: Monster (Wait)
	: ◇ Jump: +0, +1
◆Wait	: 60 frames
◆Play SE	: Monster3 (30, 100, 0)

◆Wait : 60 frames

◆Set Movement Route : Monster (Wait)
: ◆ Move toward Player

Setelah itu tambahkan *event Battle Processing* dan atur seperti gambar di bawah ini.



Gambar 3.145. Menambahkan *event Battle Processing*



Gambar 3.146. Tampilan *event Battle Processing*

Setelah menambahkan *event Battle Processing* maka akan muncul dua kondisi, yaitu *If Won* dan *If Lose*. Tambahkan *event* di bawah baris *If Won* dengan *event* di bawah ini.

◆Wait : 60 frames

◆Text : **Monster(7), Window, Bottom**

: \n<Darklord>TIDAAAK...

◆Play SE : Fire3 (30, 100, 0)

◆Set Movement Route : **Monster (Wait)**

: ◆Transparent ON

Control Switches: #0007 Monster kalah = ON

◆Wait : 120 frames

◆Text : **Actor1(1), Window, Bottom**

: \n<Kenta>Akhirnya kalah juga monster itu.

◆Text : **Actor3(7), Window, Bottom**

:\n<Izzi>Berarti benar\.\., kita emang orang yang ada di ramalan itu.

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>Nggak terlalu sulit ternyata.

◆Wait : 120 frames

◆Text :Actor3(7), Window, Bottom

:\n<Izzi>Sekarang kita ngapain?

◆Text :Actor1(1), Window, Bottom

:\n<Kenta>Kita pulang.

◆Text :Actor2(6), Window, Bottom

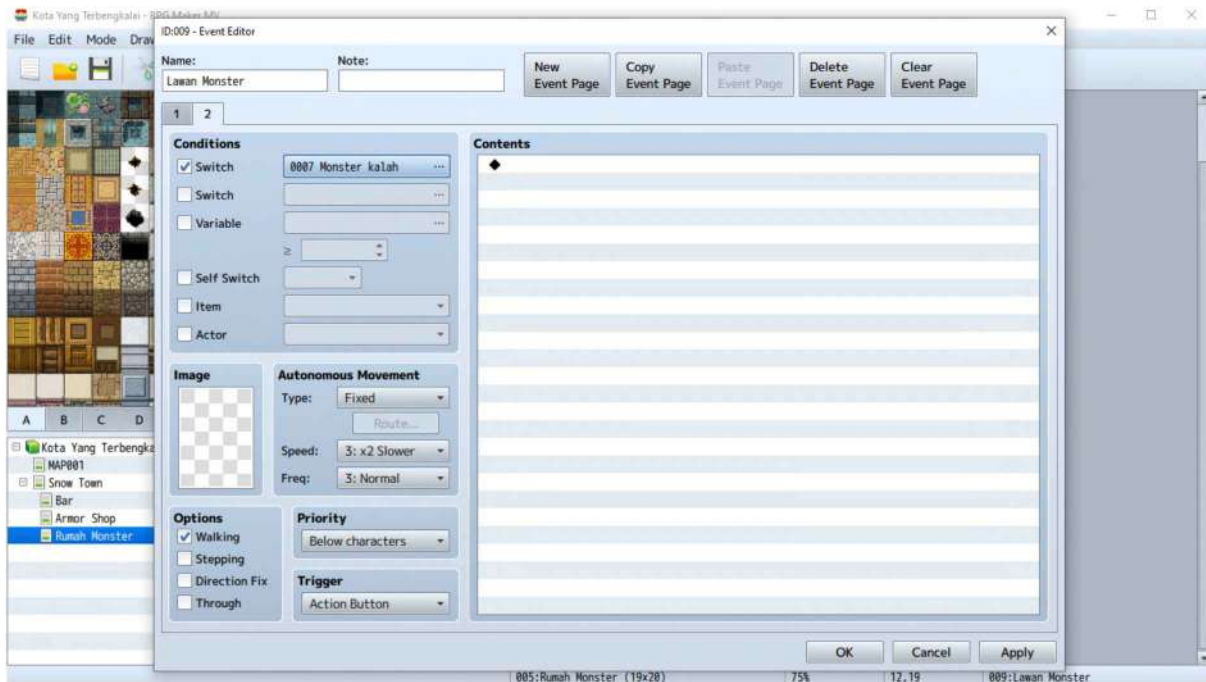
:\n<Dicky>Ayo kita keluar.

Kemudian tambahkan *event Game Over* di bawah baris *If Lose*.



Gambar 3.147. Tampilan *Game Over*

Setelah itu klik halaman dua lalu centang *Switch 007 Monster Kalah* dan kosongkan *event* di kolom *Contents*.

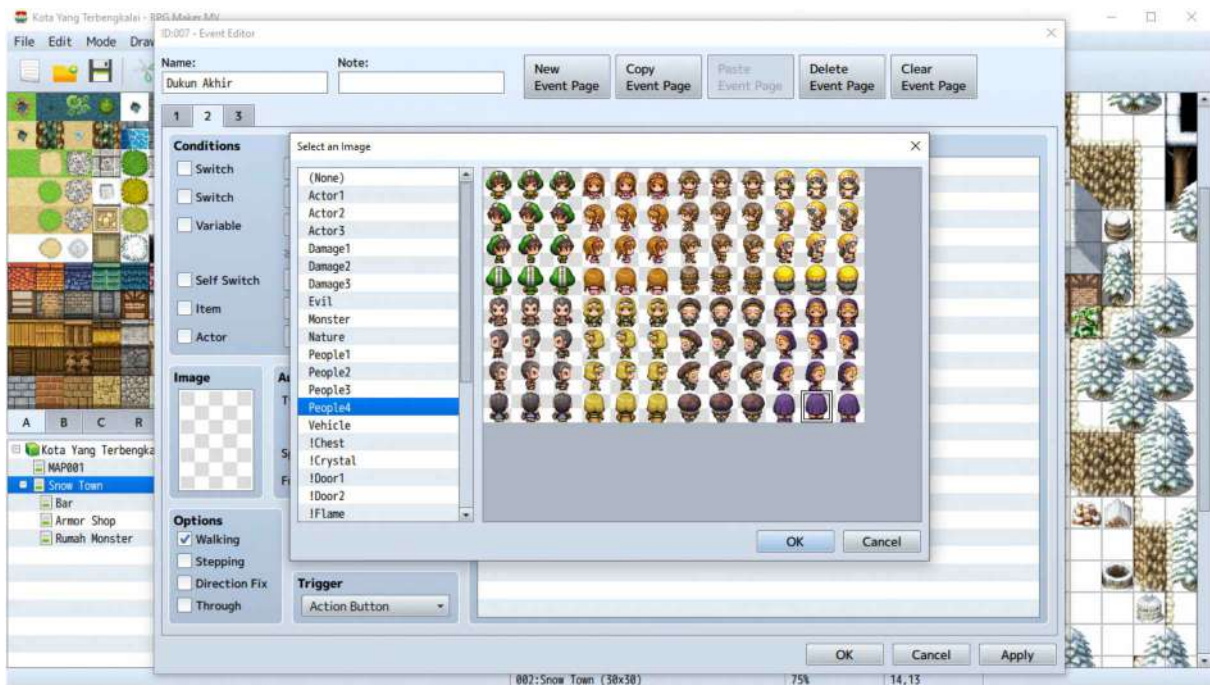


Gambar 3.148. Tampilan halaman kedua *event* Lawan *Monster*

Adekan Akhir

Sebelum masuk ke *credit*, ketiga karakter akan bertemu dengan Dukun setelah pemain keluar dari Rumah *Monster*. Klik *map Snow Town* di sebelah kiri sampai *map* berganti tampilan menjadi *map Snow Town*.

Setelah itu klik dua kali pada koordinat (14,13) lalu beri nama *eventnya* dengan nama **Dukun Akhir**. Klik *Copy Event Page* dan *Paste Event Page* dua kali sehingga terdapat tiga halaman *Event Editor*. Biarkan halaman satu secara *default* lalu klik halaman dua kemudian klik kolom *Image* dan pilih seperti gambar di bawah ini.



Gambar 3.149. Menambahkan *NPC* Dukun Akhir

Centang *Switch 007 Monster* kalah dan ubah *Triggernya* menjadi *Autorun*. Setelah itu tambahkan *event* seperti tabel di bawah ini.

◆Set Movement Route : Player (**Wait**)

: ◇ Move Down

◇ Move Down

◇ Move Down

◇ Move Down

◇ Move Down

◆Wait :120 frames

◆Text : **People4(7), Window, Bottom**

: \n<Dukun>Terima kasih nak, karena kalian ramalanku benar.

◆Text : **Actor1(1), Window, Bottom**

: \n<Kenta>Sama-sama nek.

◆Text : **People4(7), Window, Bottom**

\n<Dukun>Sekarang kota ini telah aman. Kalian boleh tinggal di sini kalau kalian mau.

◆Text : **Actor2(6), Window, Bottom**

: \n<Dicky>Terima kasih nek, tapi kami harus pulang.

◆Text : **Actor3(7), Window, Bottom**

: \n<Izzi>Iya nek, orang tua kami pasti marah karena kami pergi terlalu jauh.

◆Text : **People4(7), Window, Bottom**

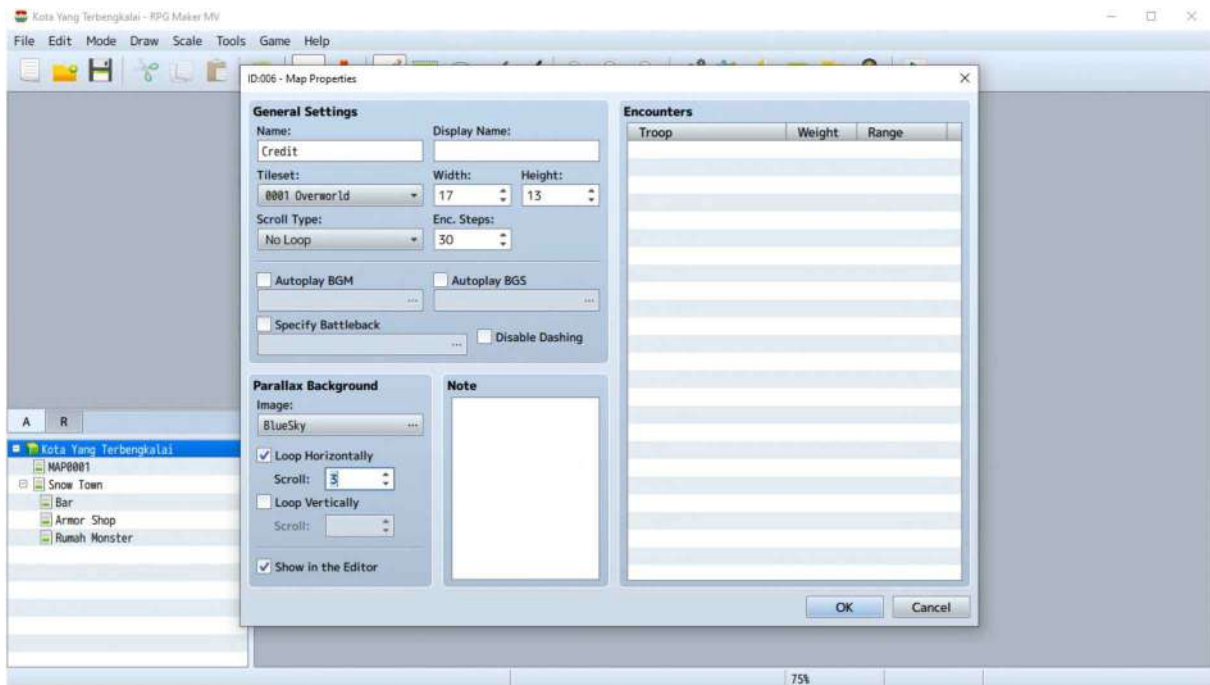
: \n<Dukun>Baiklah nak, hati-hati di jalan.

Fadeeout Screen

◆Set Movement Route : Player (**Wait**)

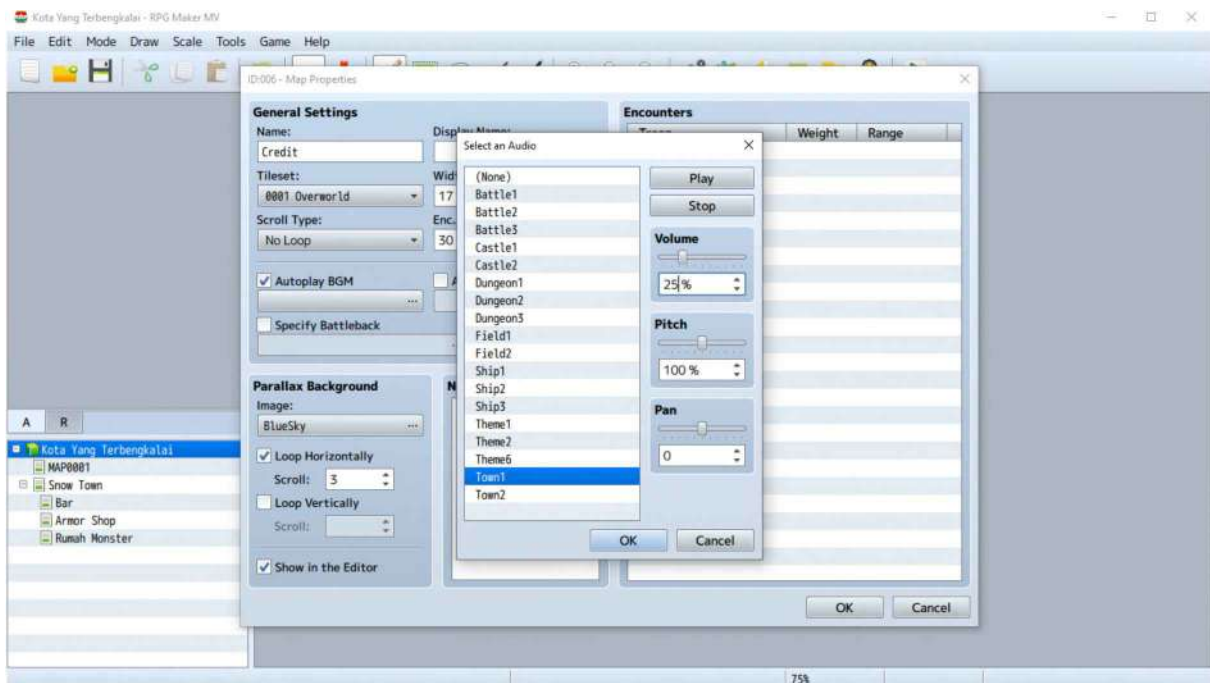
: ◆Transparent ON

Langkah selanjutnya adalah membuat *Credit* dengan cara klik kanan pada menu yang tertulis judul *game*, kemudian klik *New*.



Gambar 3.150. Membuat *map Credit*

Setelah itu atur *Autoplay BGM* seperti ini



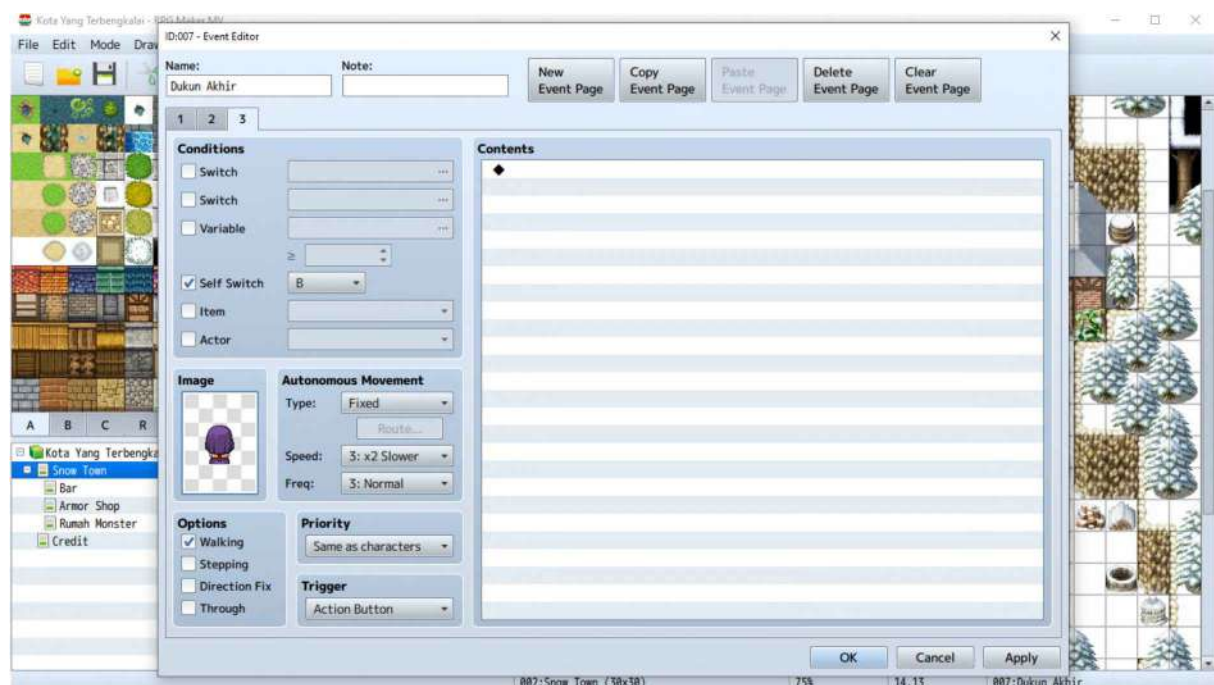
Gambar 3.151. Mengatur *Autoplay BGM* map *Credit*

Hapus saja MAP0001 setelah itu klik kembali *map Snow Town* di sebelah kiri lalu buka kembali *event* Dukun Akhir halaman dua kemudian tambahkan *event* di bawah ini.

◆Transfer Player :Credit (1,3)

Control Self Switch B = ON

Selanjutnya klik halaman tiga *event* Dukun Akhir kemudian atur seperti ini.

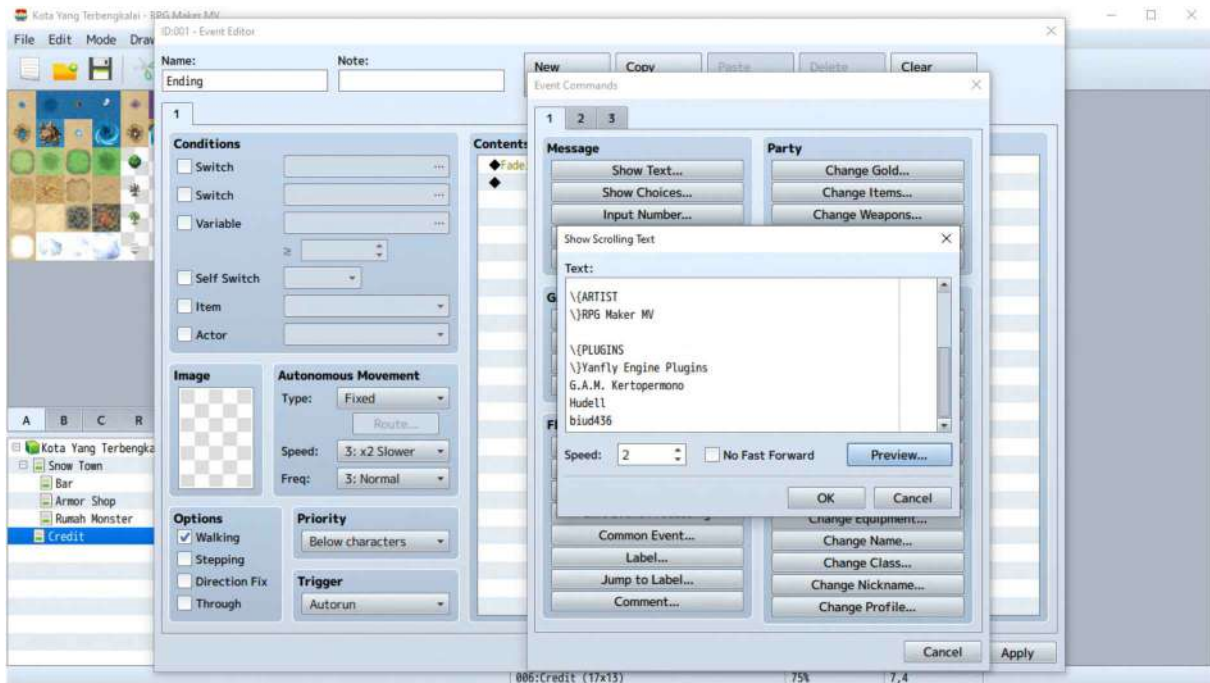


Gambar 3.152. Halaman tiga *event* Dukun Akhir

Membuat Credit

Klik *map Credit* di sebelah kiri sampai *map* kemudian klik di koordinat bebas untuk membuat *credit*. Ganti *Trigger*-nya

menjadi *Autorun* kemudian masukkan *event Fadein Screen* lalu tambahkan *event Show Scrolling Text*.



Gambar 3.153. Membuat *Show Scrolling Text*

Tulis *Show Scrolling Text* seperti ini.

```
\\TA[1]\\{DEVELOPER  
\\}Nama anda
```

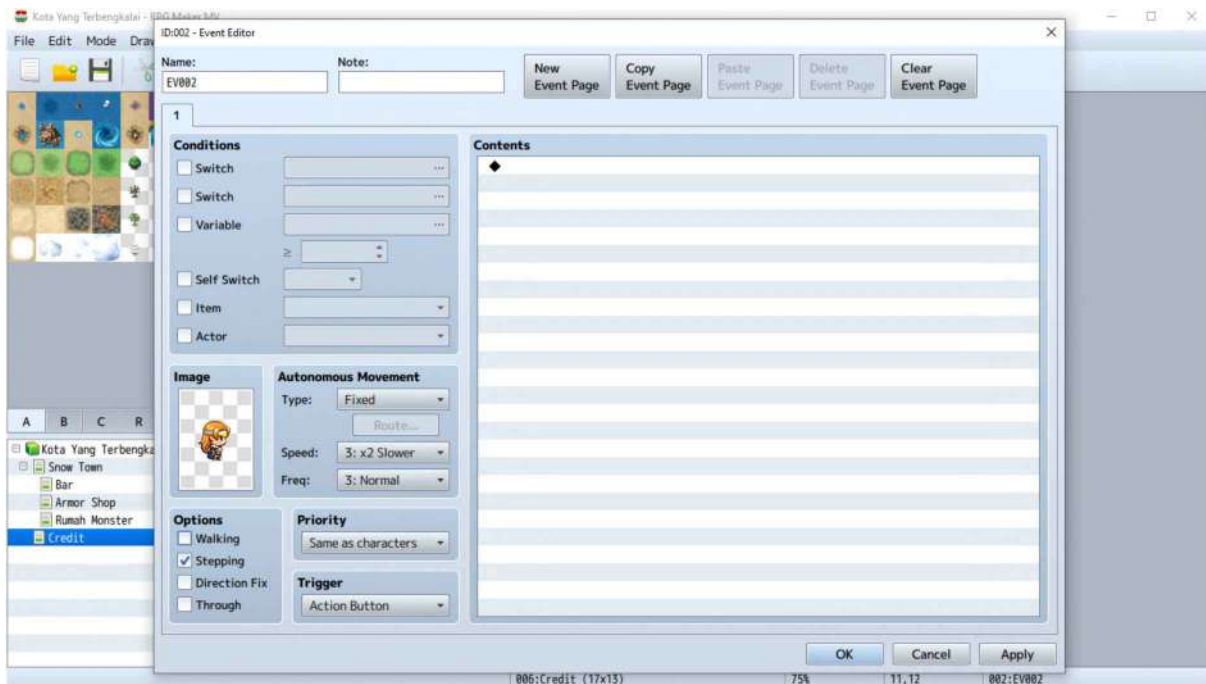
```
\\{MUSIC DESIGNER  
\\}RPG Maker MV
```

```
\\{ARTIST  
\\}RPG Maker MV
```

Kode \\{ memiliki arti menambah ukuran font dan kode \\} untuk mengurangi ukuran font.

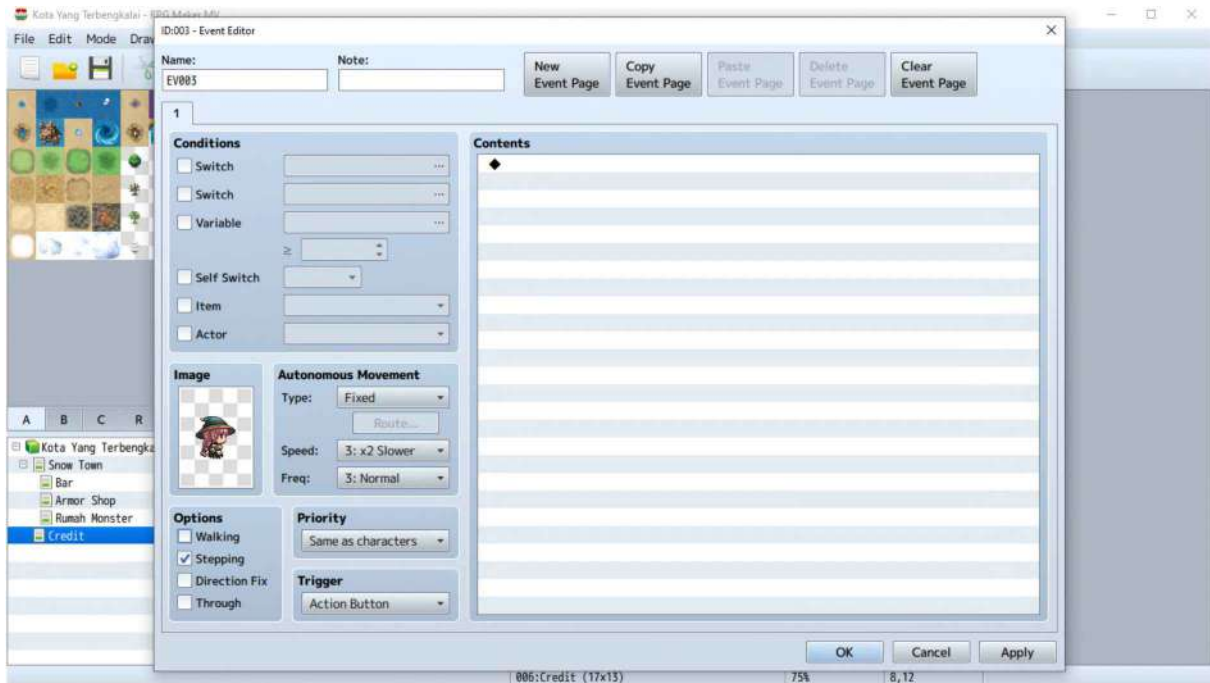
```
\\{PLUGINS  
\\}Yanfly Engine Plugins  
G.A.M. Kertopermono  
Hudell  
biud436
```

Setelah itu tambahkan *event Wait: 120 frames* dan *event Return to Title Screen*. Kemudian klik dua kali pada koordinat (11,12) untuk membuat karakter Kenta dan buat menjadi seperti ini.



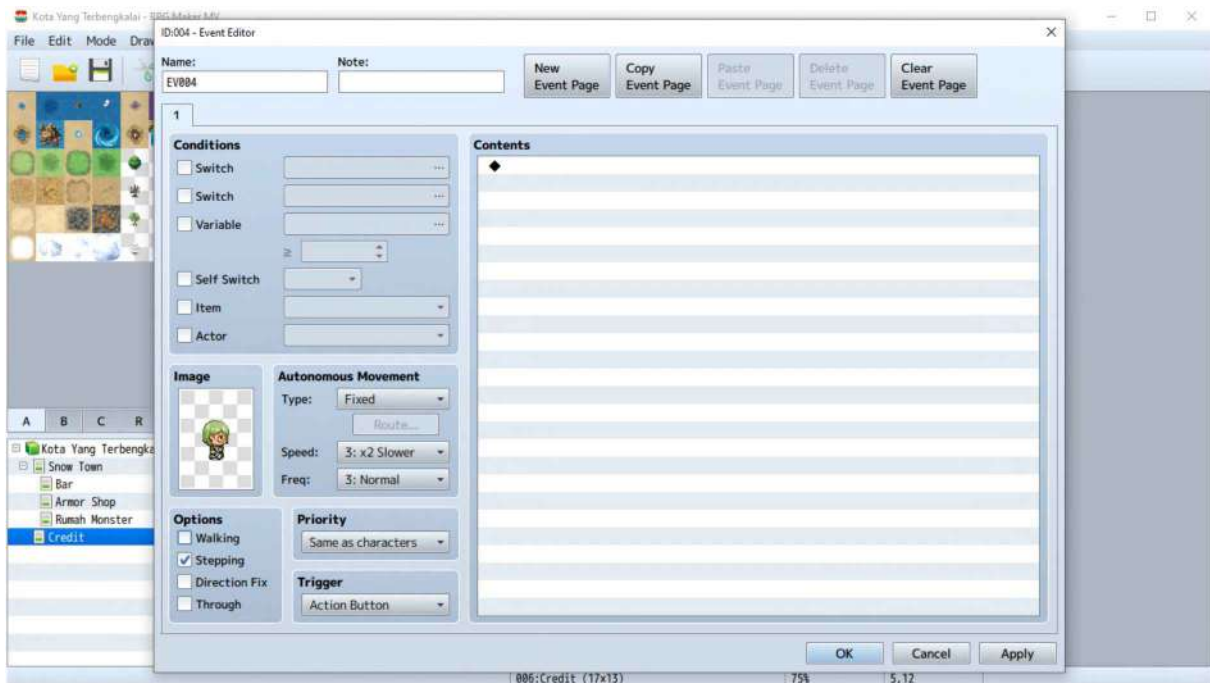
Gambar 3.154. *Credit* karakter Kenta

Setelah itu klik dua kali pada koordinat (8,12) untuk karakter Izzi dan buat menjadi seperti ini.



Gambar 3.155. *Credit* karakter Izzi

Yang terakhir buat *event* untuk karakter Dicky di koordinat 5,12 seperti ini.



Gambar 3.156. *Credit* karakter Dicky

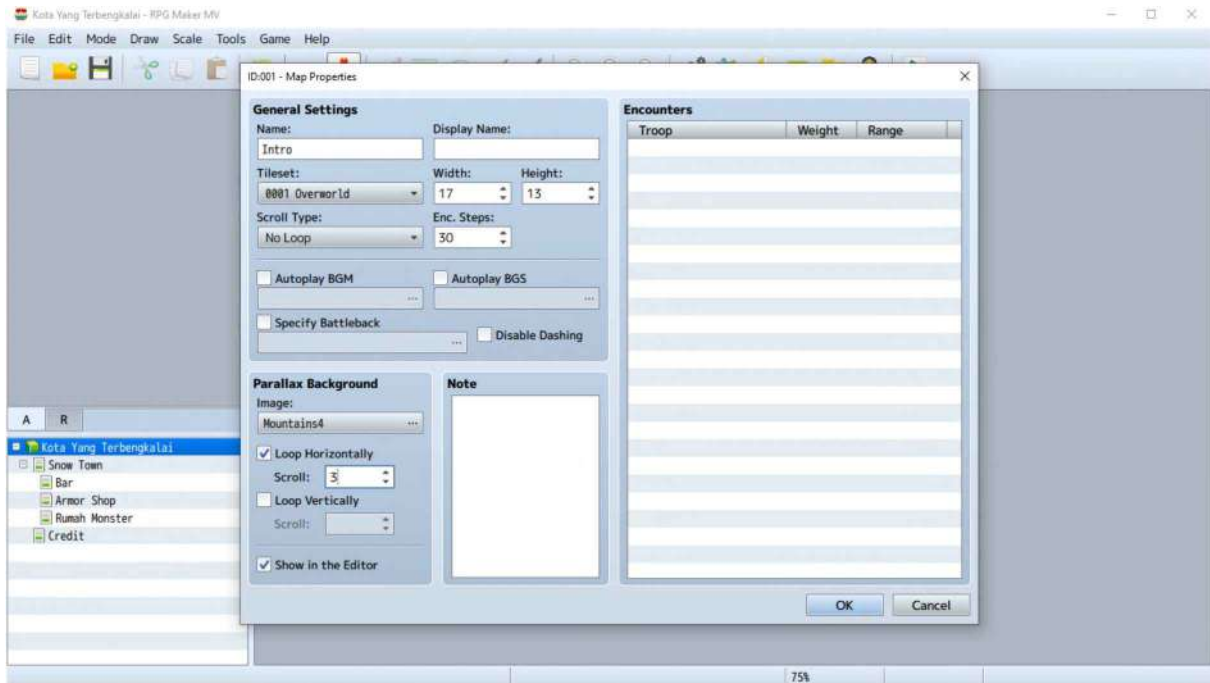
Hasilnya akan seperti ini.



Gambar 3.157. Tampilan *Credit*

Membuat Intro

Klik kanan pada menu yang tertulis judul *game*, kemudian klik *New* lalu buat seperti ini.



Gambar 3.158. Membuat *map Intro*

Setelah itu klik di koordinat bebas untuk membuat *intro*. Ganti *Triggernya* menjadi *Autorun* kemudian masukkan *event* seperti tabel di bawah ini.

◆Set Movement Route : Player (**Wait**)

: ◆Transparent ON

◆Wait : 120 frames

◆Text :None, Transparent, Middle

:\TA[1]Izzi\, Kenta\, dan Dicky.\|\ Tiga orang bocil yang tidak sengaja terdampar

di sebuah kota kecil karena mereka kecerobohan setelah bermain

terlalu jauh dari rumah. \|\|\| Orang tua mereka diperkirakan sudah

menunggu mereka sambil membawa sapu yang siap memukul mereka.

◆Wait : 60 frames

◆Text :None, Transparent, Middle

:\TA[1]Karena sudah terlanjur sampai di pulau tersebut,

mereka bertiga memulai petualangannya.\|

Dan berharap\..\..\|Bisa pulang ke rumah.

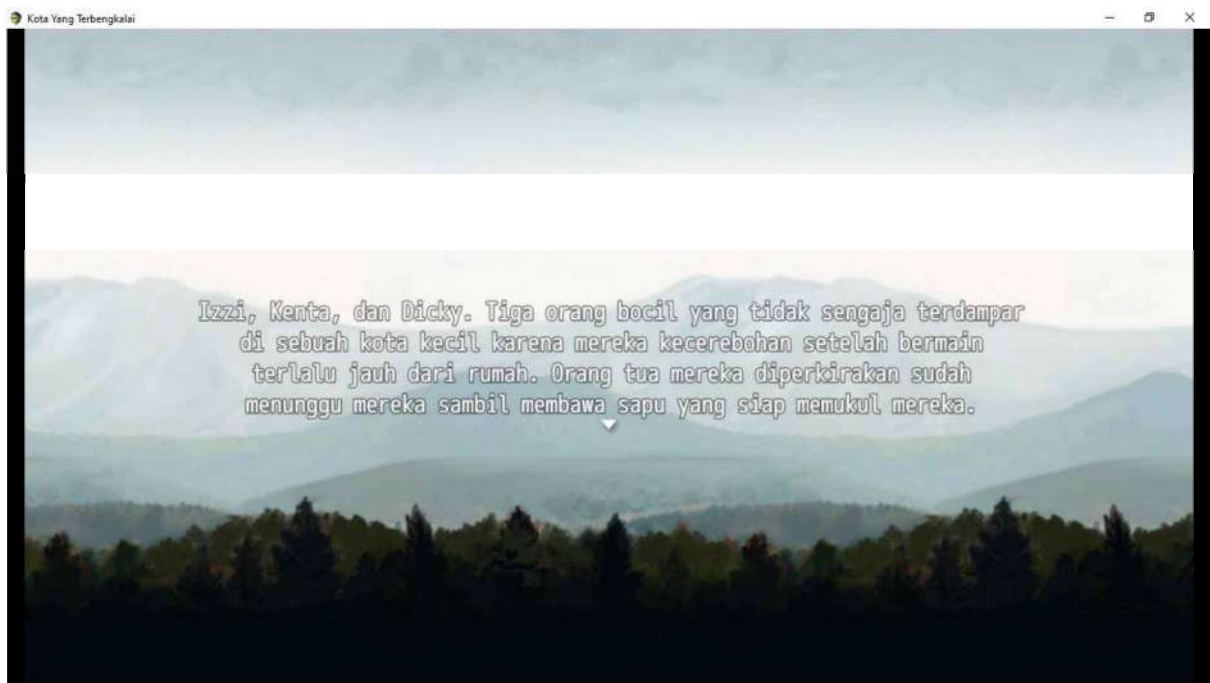
Fadeout Screen

◆Wait : 120 frames

◆Set Movement Route : Player (**Wait**)

: ◆Transparent OFF

◆Transfer Player : Snow Town (14,29) (Direction Up)



Gambar 3.159. Tampilan *map Intro*

Selanjutnya klik kanan di koordinat bebas pada *map Intro* lalu klik kanan → *Set Starting Position* → *Player* untuk memilih *map Intro* sebagai *map* pertama saat *game* dijalankan.

Membuat Event Tambahan

Snow Town

Klik dua kali pada koordinat (14,25) di *map Snow Town* lalu ubah *Priority*-nya menjadi *Same as Characters*. Setelah itu tambahkan *Conditional Branch If: Player is Facing Up* di baris *Contents* dan centang *Create Else Branch*. Kemudian tambahkan *event* di bawah baris *If: Player is Facing Up* seperti ini.

◆Text :None, Window, Bottom

:\TA[1]Kota Marobone

:\Populasi: ?

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>Nice info gan, suruh nebak jumlah populasinya sendiri.

◆Text :Actor1(1), Window, Bottom

:\n<Kenta>Aku kira Desa Fu*king, Austria. \.\.Eh Fugging.

◆Text :Actor3(7), Window, Bottom

:\n<Izzi>Kukira Wakanda\.\, salah satu negara penghasil kelapa sawit tapi harga

:\minyaknya mahal.

Setelah itu tambahkan *event* di bawah baris *Else*

◆Text :Actor2(6), Window, Bottom

:\n<Dicky>Nggak kelihatan kalau bacanya dari sini.

Selanjutnya klik dua kali pada koordinat (14,21) lalu ubah *Priority*-nya menjadi *Same as Characters*. Setelah itu

tambahkan *Conditional Branch If: Player is Facing Up* di baris *Contents* dan centang *Create Else Branch*. Kemudian tambahkan *event* di bawah baris *If: Player is Facing Up* seperti ini.

◆Text :None, Window, Bottom

: \TA[1]Kiri: Bar

: Kanan: Penginapan

: JANGAN LURUS

Setelah itu tambahkan *event* di bawah baris *Else*

◆Text :Actor2(6), Window, Bottom

: \n<Dicky>Nggak kelihatan kalau bacanya dari sini.

Selanjutnya klik dua kali pada koordinat (25,11) lalu ubah *Priority*-nya menjadi *Same as Characters*. Setelah itu Klik *Copy Event Page* dan *Paste Event Page* sehingga terdapat dua halaman *Event Editor*. Kemudian tambahkan *event* di halaman satu seperti ini.

◆Text :Actor1(1), Window, Bottom

: \n<Kenta>Temboknya udah retak, hancurin sekalian yok.

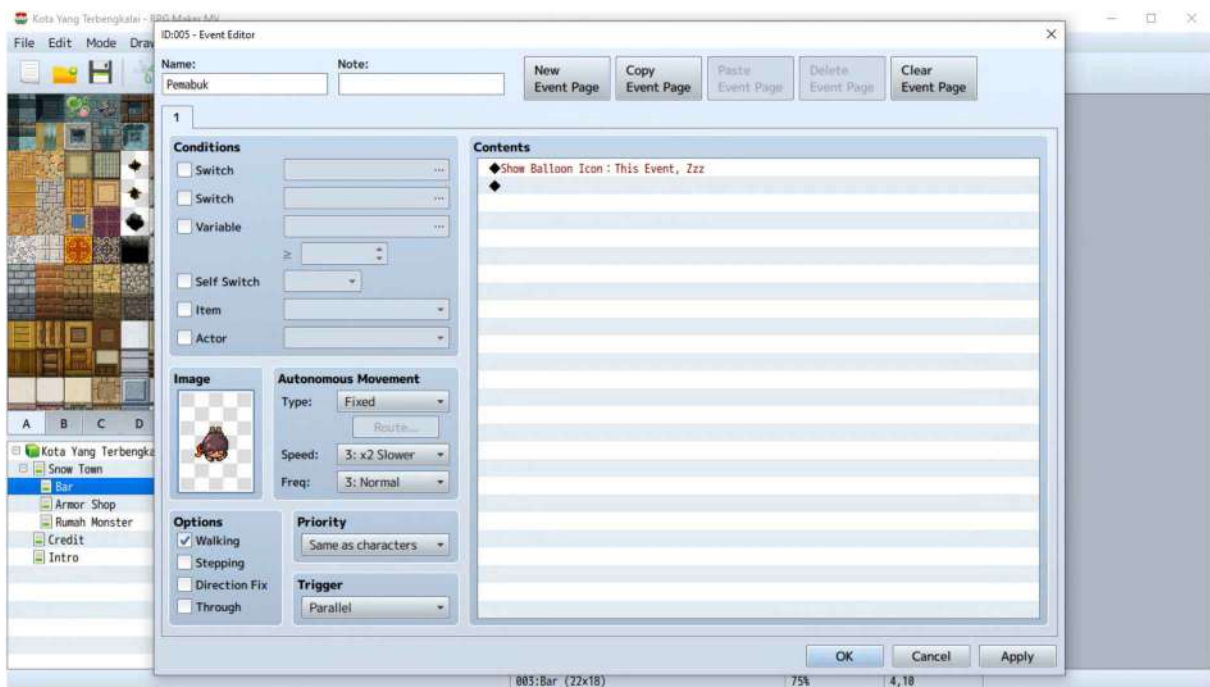
◆Text :Actor3(7), Window, Bottom

: \n<Izzi>Kita baru aja dateng lho, udah main vandalisme aja.

Setelah itu klik halaman dua kemudian centang *Switch 0001 Mencari Informasi* dan biarkan kolom *Contents* kosong.

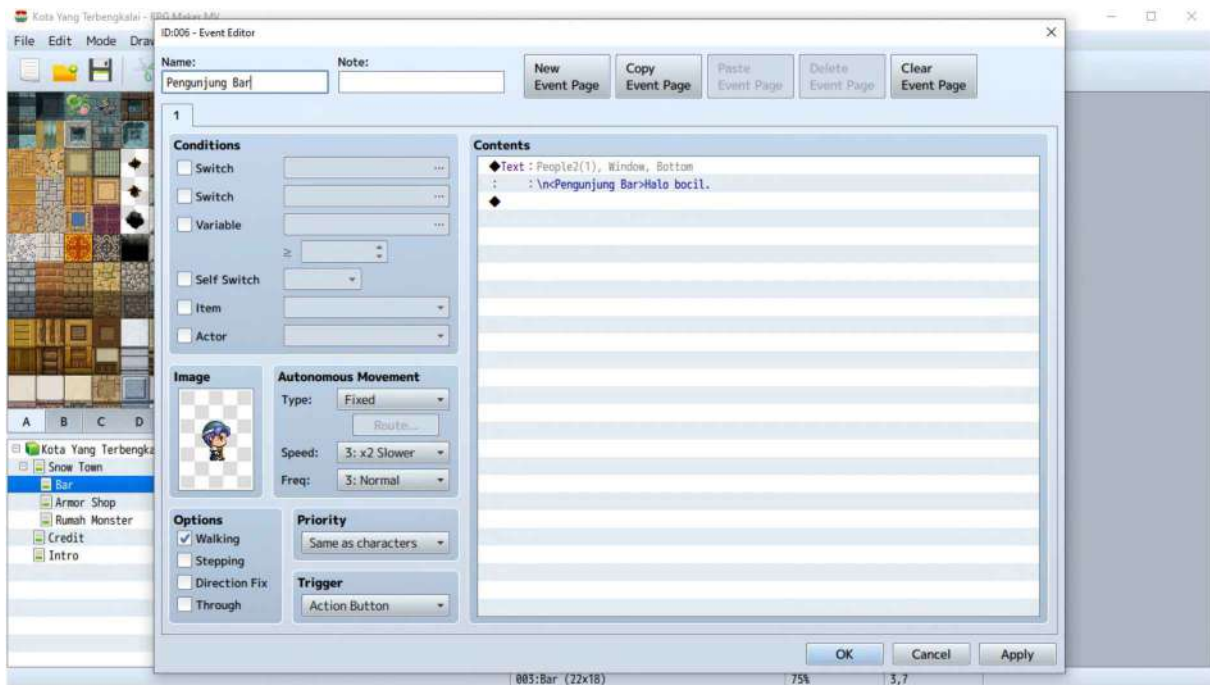
Bar

Klik dua kali pada koordinat (4,10) di *map Bar* lalu ubah *Triggernya* menjadi *Parallel*. Kemudian tambahkan *event* di seperti ini.



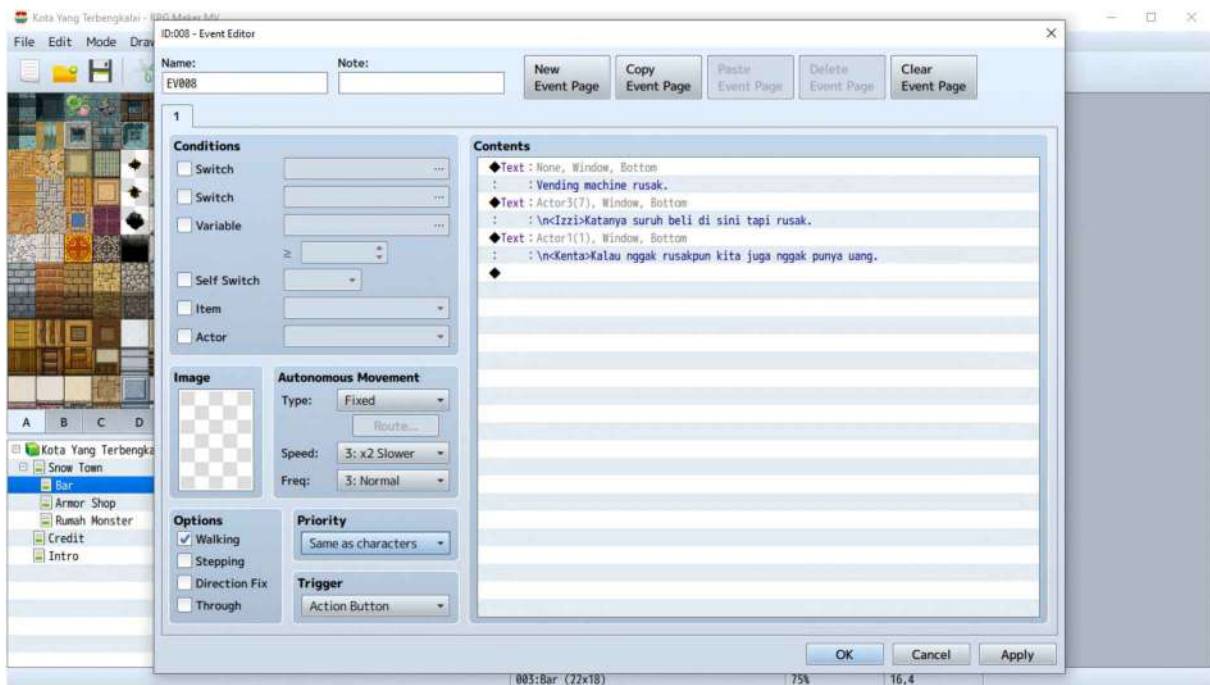
Gambar 3.160. *Event* tambahan Pemabuk

Selanjutnya klik dua kali pada koordinat (3,7) kemudian tambahkan *event* di seperti ini.



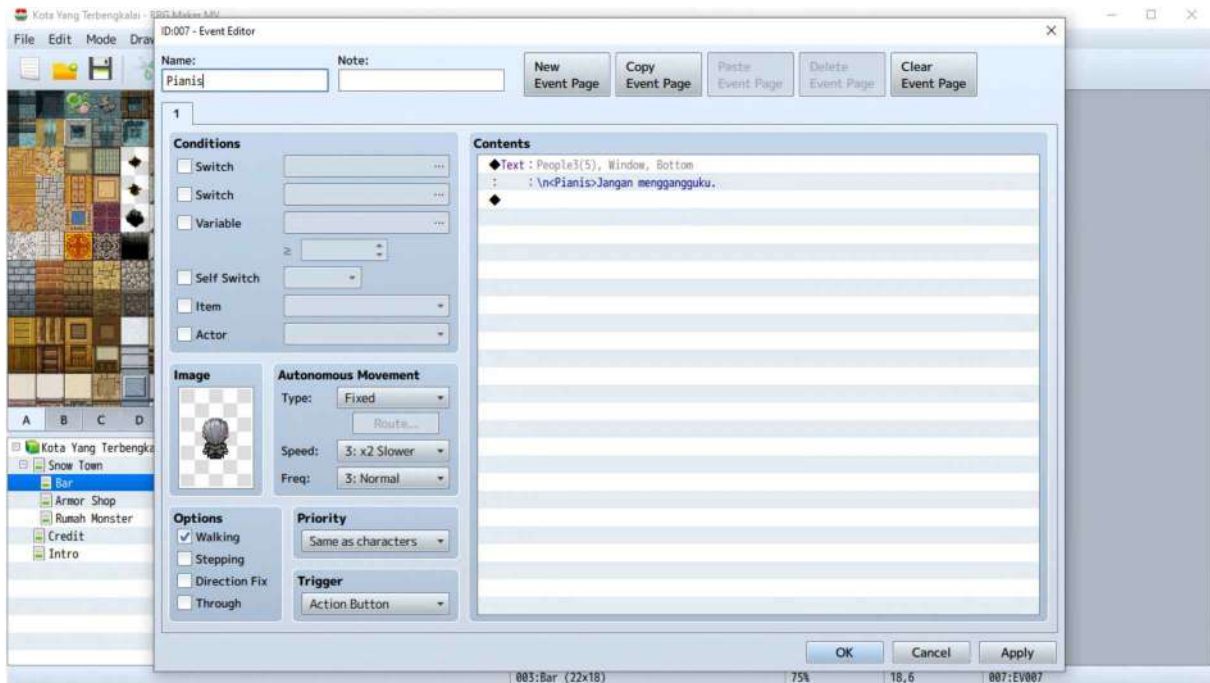
Gambar 3.161. *Event* tambahan Pengunjung Bar

Kemudian klik dua kali pada koordinat (16,4) kemudian tambahkan *event* di seperti ini.



Gambar 3.162. *Event* tambahan Vending Machine

Yang terakhir klik dua kali pada koordinat (18,6) kemudian tambahkan *event* di seperti ini.



Gambar 3.163. *Event* tambahan Pianis

Hasil Akhir

Untuk hasil akhir dari *game* ini bisa dilihat di <https://s.id/KotaYangTerbangkalai>.

CARI JALAN KELUAR

Kristoforus Thoeirick Dimasaputra



Gambar 4.1. *Game Icon*

Analisa Game

Cari Jalan Keluar adalah permainan *escape room* dengan unsur *survival*, yang dimana pemain harus menemukan cara untuk keluar dari sebuah ruangan selagi menghindari kejaran *zombie*.

Mission Statement

Cari Jalan Keluar merupakan permainan *Role-Playing Game*, *Survival*, dan *Escape Room*, dimana pemain harus mencari kunci untuk keluar dari ruangan selagi mempertahankan diri dari *zombie* yang mengejar.

Genre

Role-Playing Game, *Survival*, dan *Escape Room*.

Platform

PC atau *Personal Computer*

Target Audience

Permainan ini ditargetkan kepada orang-orang yang menyukai permainan dengan genre *survival* dan *escape room*.

Gambaran Umum Gameplay

Permainan single player Cari Jalan Keluar memiliki genre *Role-Playing Game*, *Survival*, dan *Escape Room*. Pemain dapat berinteraksi dengan benda-benda untuk menemukan petunjuk membuka pintu keluar menuju *level* selanjutnya.

Pada ruangan ketiga atau *level* tiga, pemain akan bertemu dengan *zombie* yang akan mengejar dan memberi luka kepada pemain. Pada saat ini pemain harus mencari petunjuk membuka pintu sekaligus menghindari kejaran *zombie*.

Pengalaman Pemain

Kau terbangun di sebuah ruangan yang mirip seperti penjara. kau berhasil keluar dan mendapatkan kunci ke ruangan selanjutnya dan bertemu dengan *zombie* yang mengejarmu. Tugasmu adalah mencari kunci menuju pintu keluar selagi bertahan hidup dan menghindari *zombie*.

Panduan Gameplay

Permainan ini dianjurkan untuk dimainkan oleh remaja tiga belas tahun keatas karena ada unsur kekerasan.

Mekanik Gameplay

Pemain dapat menggerakkan *avatar* kearah atas (↑), kiri (←), bawah (↓), dan kanan (→). Terdapat juga *system* untuk mengambil dan menyimpan benda kedalam tas. Tingkat kesulitan *level* akan naik setiap menyelesaikan satu ruangan.



Gambar 4.1. *Character*

Lise (MC)

Tipe: Manusia



Gambar 4.2. *Lise*

Sifat: *Passive dan Stubborn*

Info Manusia

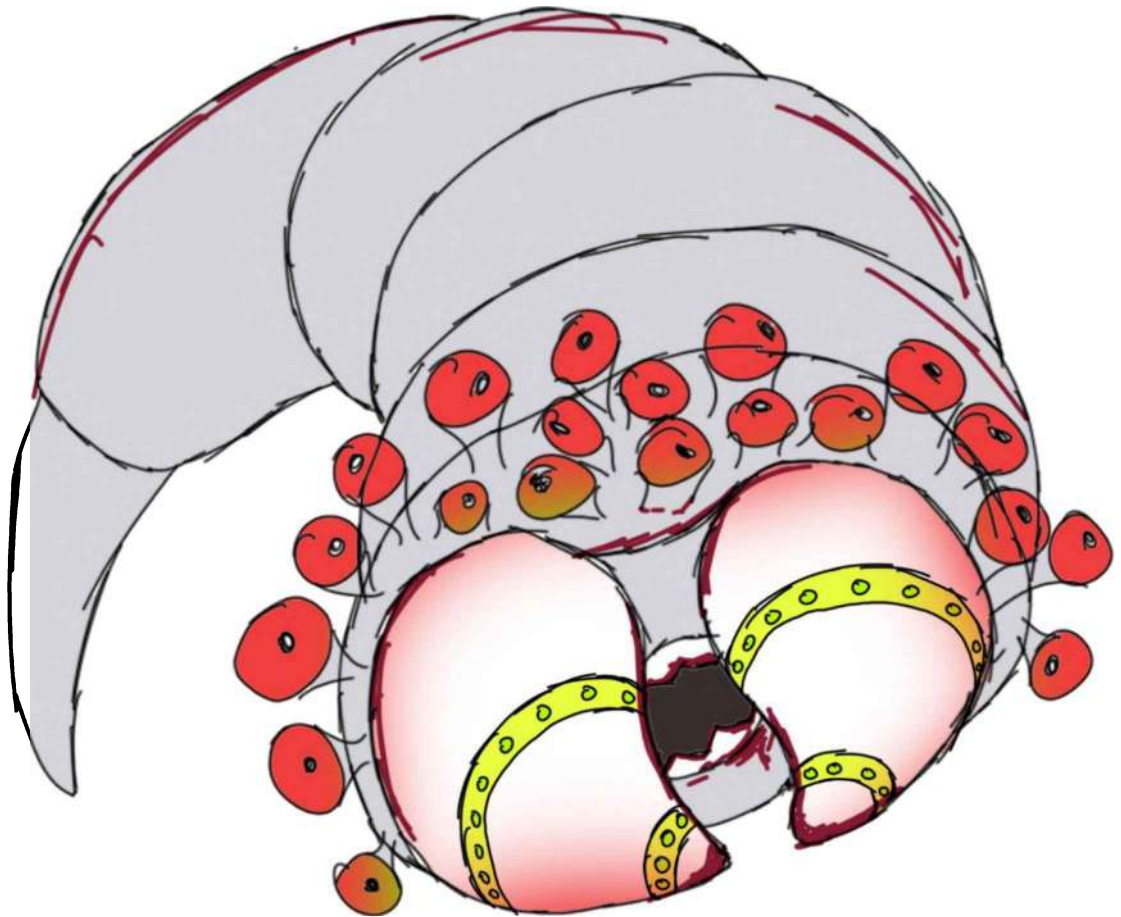
Semua pemerintah di seluruh dunia mengungsikan rakyatnya ke dalam *bunker* bom untuk berlindung dan mengurangi jumlah *Zombie*.

Di dalam *bunker* terdapat berbagai fasilitas untuk mendukung keberlangsungan hidup manusia tanpa perlu keluar dari bunker. Terdapat fasilitas seperti rumah sakit, sekolah, kebun, pusat kebugaran, dan lain lain.

Ada pula pasukan tentara yang melakukan ekspedisi untuk mengurangi jumlah *zombie* sekaligus membawa manusia yang berhasil bertahan di luar *bunker*.

Parasitic Worm

Tipe: *Parasit*



Gambar 4.3. *Parasitic Worm*

Sifat: *Aggressive dan Violent*

Parasitic Worm atau cacing parasit adalah organisme asing yang pertama kali ditemukan di kawah yang terbentuk oleh meteoroid yang mendarat di bumi. Cacing parasit ini dapat menyerang makhluk hidup khususnya mamalia, dan menyebabkan mutasi pada tubuh inangnya.

Manusia yang terinfeksi akan mengalami gejala demam tinggi yang disusul dengan kejang-kejang. Pada masa ini cacing parasit akan menyerang otak untuk mengambil alih tubuh inangnya. Setelah berhasil, pada punggung tubuh inang akan muncul benda seperti kepompong yang menandakan mutasi telah dimulai.

Mutasi ini dilakukan untuk meningkatkan kemampuan bertahan hidup tubuh inang yang dikendalikan oleh cacing parasit.

Zombie I (ZI)

Tipe: *Zombie*



Gambar 4.4. *Zombie I (ZI)*

Sifat: *Passive dan Timid*

Zombie II (ZII)

Tipe: *Zombie*



Gambar 4.5. *Zombie II (ZI)*

Sifat: *Aggressive dan Violent*

Zombie adalah manusia yang terinfeksi oleh cacing parasit dan telah selesai bermutasi. Pada tubuh *zombie* ada benda seperti ekor yang berfungsi sebagai bentuk pertahanan diri serta untuk berburu mangsa. Sejauh ini ada dua jenis *Zombie* yang telah ditemukan.



Gambar 4.6.
Ekor ZI

Jenis pertama dengan kode nama ZI, memiliki ekor dengan alat penyengat yang kuat dan beracun. ZI berburu dengan bersembunyi di tempat gelap dan akan menyuntikan racun yang akan membuat tubuh mangsanya lumpuh.

Saat merasa terpojok ZI akan melompat dan menyerang menggunakan penyengatnya. Berdasarkan hasil penelitian, penyengat ZI dapat menembus pelindung setebal 30mm.



Gambar 4.7. Ekor Capit

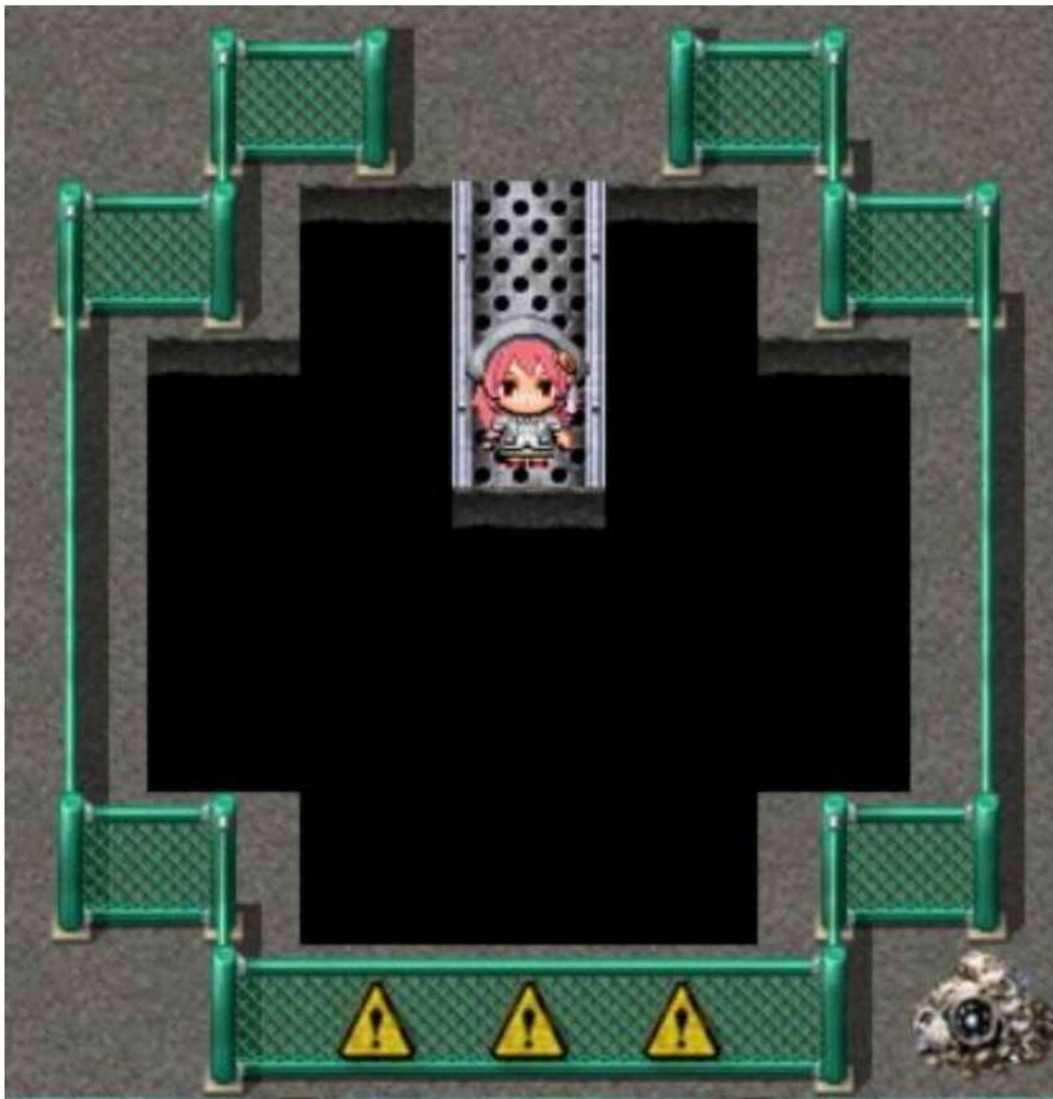
Zombie kedua memiliki kode nama ZII, memiliki dua jenis ekor. Salah satunya memiliki bentuk menyerupai penjepit yang digunakan untuk menangkap dan melumpuhkan mangsanya, ekor kedua ZII berbentuk seperti tentakel dengan mulut penghisap dilengkapi gigi-gigi kecil tajam di ujungnya.



*Gambar 4.8. Ekor
Tentakel*

Zll berburu dengan menggunakan ekornya sebagai perangkap. Setelah mangsa dilumpuhkan oleh ekor penjepit, bagian ujung ekor tentakel akan terbuka dan kemudian mulut yang dipenuhi gigi akan membuat luka untuk menghisap darah mangsanya.

Desain Level



Gambar 4.9. *Level Map*

Akan tersedia lima *level*, dua *level* awal merupakan tingkat tutorial. *Level* tiga sampai dengan lima, akan ada *zombie* yang mengejar pemain.



Gambar 4.10. *Ruang pertama*

Ruangan tanpa jendela dan pintu tanpa gagang. Ini adalah ruangan tempat pemain pertama kali berada, di map ini pemain dapat berinteraksi dengan benda-benda untuk mencari dan menemukan cara untuk keluar dari ruangan tersebut.

Level 2

Infirmary

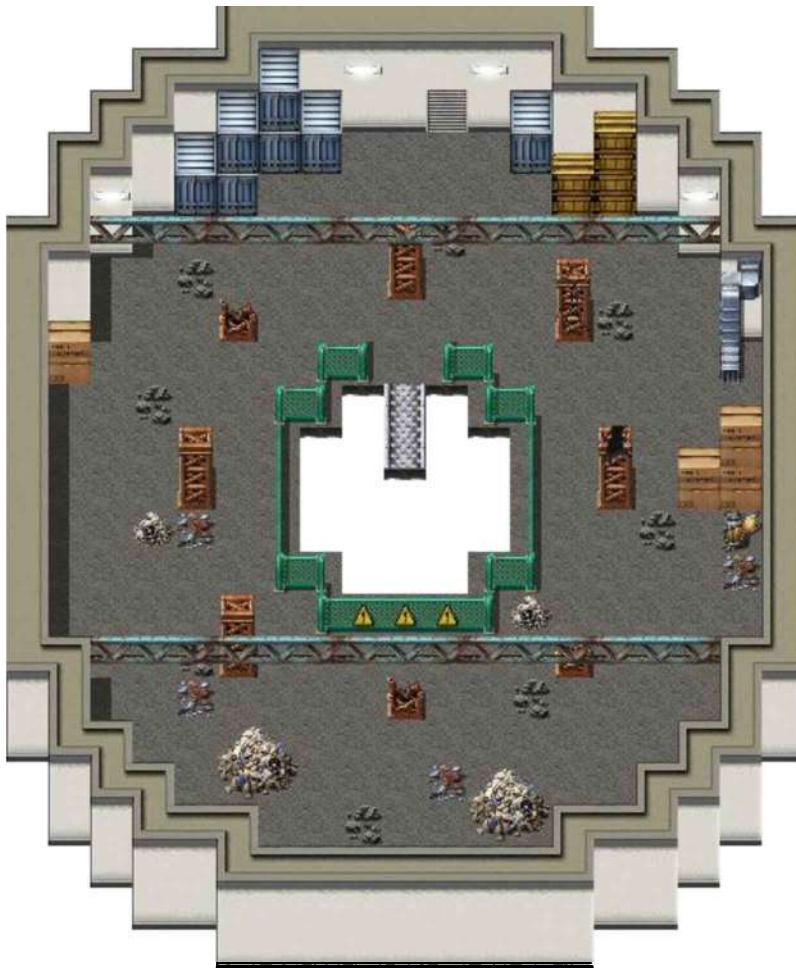


Gambar 4.11. *Ruang Kedua*

Rumah sakit yang terbengkalai. Pemain harus mencari sebuah kunci yang berada di salah satu ruangan untuk membuka pintu keluar.

Level 3

Storage Room



Gambar 4.12. *Ruang Ketiga*

Pada ruangan penyimpanan ini, tidak ada kunci yang perlu ditemukan ataupun pintu untuk dibuka, pada ruangan ini akan ada *Zombie* yang bersembunyi di kegelapan.

Level 4

Lobi



Gambar 4.13. *Ruang Keempat*

Tempat santai para penghuni *bunker*, alihkan perhatian *zombie* agar dapat menuju ke pintu selanjutnya.

Level 5

Pintu Keluar



Gambar 4.14. *Ruang Kelima*

Ruangan terakhir menuju kebebasan, pintu keluar *bunker* di jaga oleh tim ekspedisi dan rakyat sipil tidak memiliki izin.

Jalan cerita

Aku terbangun di sebuah ruangan yang tidak kukenali, ruangan tanpa jendela dan sebuah pintu tanpa gagang. Hanya diriku, sebuah ranjang besi, dan meja kayu kecil yang ada di ruangan itu.

“Mengapa aku berada di ruangan seperti penjara ini?” Aku bergumam.

Aku menemukan sebuah ventilasi kecil yang dapat menjadi jalan keluar ku. Aku masuk ke dalam ventilasi dan keluar di koridor rumah sakit yang terbengkalai, banyak tempat tidur pasien rusak dan botol-botol obat berserakan.

Terdapat tiga pintu keluar dari ruangan ini namun dua pintu terhalang oleh reruntuhan dan tidak dapat kulewati, sehingga hanya ada satu pintu yang menjadi pilihan ku. Pintu itu terkunci dan memiliki label A39.

“Kurasa itu harus menemukan kunci dengan label A39.” Aku bergumam kesal.

Aku menemukan kotak tempat menyimpan kunci di salah satu ruangan. Setelah mendapatkan kunci A39, Aku langsung

bergegas untuk membuka pintu keluar. Dan yang aku melihat satu makhluk sedang berjalan di ujung ruangan.

Makhluk itu mirip seperti manusia namun memiliki ekor, dan makhluk itu terlihat seperti sedang menyeret sebuah kaki dan membuat jejak darah.

Aku mengendap-endap masuk kedalam ruangan yang berbentuk lingkaran yang ditengahnya terdapat, lubang cukup dalam sampai aku tidak dapat melihat dasarnya

“Klang!!” Suara kaleng yang tidak sengaja ku tendang.

Zombie yang ada di ruangan itu langsung menyadari keberadaan ku, dan mulai mengejar. Aku pun ikut berlari mengelilingi ruangan lingkaran ini.

Aku telah berlari mengelilingi ruangan ini namun tidak menemukan pintu selain pintu tempat ku masuk, sedangkan masih ada *zombie* yang masih mengejarku.

Aku memberanikan diri untuk kemudian melompat ke lubang yang berada di tengah ruangan ini.

“Sekarang atau tidak sama sekali.” Aku berteriak selagi melompat ke dalam lubang.

“Splash.” Suara air tempat ku mendarat, aku telah jatuh cukup dalam dan aku mendarat di kolam yang cukup dalam.

Ruangan kali ini lumayan luas namun aku juga dapat mendengar suara langkah kaki *zombie* di dekat ku. Suara air saat aku terjatuh pasti menarik perhatian mereka.

Aku berkeliling untuk mencari pintu keluar selagi menghindari *zombie*, hanya ada satu pintu yang ku temukan namun ada kerumunan *zombie* yang berada di depan pintu itu

Aku Kembali ke kolam tempat ku jatuh dan melihat satu *zombie* yang suara Langkah kakinya ku dengar. Saat ini dia berada di sebelah kolam, aku mengendap-endap dan mendorongnya masuk ke dalam kolam itu.

Zombie itu membuat cukup keributan dan berhasil menarik perhatian seluruh kerumunan *zombie*. Pada saat itu aku langsung bergegas melewati pintu ke ruangan selanjutnya.

Ruangan kali ini mirip seperti sebuah hanggar pesawat dengan banyak alat-alat seperti komputer dan alat yang tidak ku ketahui. Tidak ada siapapun disini, tidak ada *zombie* maupun orang lain.

Aku melihat sebuah tombol merah dan kemudian menekannya. Suara gerigi besi yang berdecit mulai terdengar dan pintu hanggar mulai terbuka.

Aku dapat melihat rerumputan dan pepohonan hijau, sudah lama aku tidak melihat nya. Aku melangkah perlahan keluar hanggar dan mendengar suara helikopter bersiap mendarat di depan hanggar.

Suara tembakan terdengar dan aku melihat peluru bius di tangan ku. Tubuhku lemas dan mulai kehilangan kesadaran. Namun aku sempat mendengar suara radio yang mengatakan "Subject 052, experiment berhasil."

.End.

Tautan Game: <https://bit.ly/carijalankeluar>

Interogasi dan Edukasi

Samuel Matthew Gunawan

Dengan kemajuan zaman, teknologi juga semakin maju, salah satunya teknologi *game*. Sekarang *game* sudah diterima di masyarakat dan tidak diperlakukan seperti suatu hal yang membuang-buang waktu. Keahlian yang diperlukan dalam *game* pun juga berkembang, yang sebelumnya hanya kecepatan, akurasi, dan sejenisnya, sekarang ada *game* yang memerlukan kemampuan dalam kehidupan sehari-hari seperti kognitif.

Salah satu contoh *game* yang menggunakan kemampuan kognitif adalah *game Mystery*. Dalam buku ini saya akan menjelaskan tentang pembuatan *game* saya yang bernama Interogasi dan Edukasi yang merupakan *game* ber-*genre* campuran *Mystery*, *Puzzle*, *Visual Novel*, dan Edukasi. Untuk *game engine* yang akan digunakan adalah RPG Maker, selain merupakan *game engine* yang mudah untuk dipelajari *game engine* ini tidak memerlukan *coding* dalam pembuatan *game*.

Tantangan

Tantangan yang ada dalam game ini akan serupa dengan *quiz* dan argumen yang akan dilakukan oleh *character* saat *player* menunjukkan *item*/bukti yang benar saat bagian interogasi. Tergantung kondisi *quiz* atau penunjukan bukti, "nyawa" *player* bisa berkurang, juga ada beberapa kondisi dimana "nyawa" tidak akan berkurang. Selain pengurangan nyawa tentu saja player akan diajak untuk berpikir, mengingat, dan lain-lain dalam kasus.

Karakter



Gambar 5.1. Farrel Askha Mus

Main character dari *game* ini. Polisi pemula yang mendapatkan promosi ke detektif nasional pemula karena mampu menyelesaikan kasus yang tidak terselesaikan sebelumnya. Karena promosi yang tiba-tiba dia menjadi terkenal. Sekarang dia biasa dipanggil untuk kasus yang menemui jalan buntu.



Gambar 5.2. Aziz Tan

Polisi pemula yang menjadi *assistant* di sepanjang game. Merupakan teman lama Farrel.



Gambar 5.3. Bunda Pectora

Korban pertama, tertusuk di bagian perut. Ibu dari tersangka.



Gambar 5.4. Victor Timor

Korban kedua, terpukul benda tumpul.



Gambar 5.5. Michael Xthree

Satu-satunya saksi dalam kasus ini, merupakan penggemar ekstrim. *Motto*-nya adalah "Di mana ada hal ekstrim di situlah aku berada!".



Gambar 5.6. Zahra Mandu

Polisi yang bertugas dalam memandu *main character* saat akan berpindah tempat dalam sesi investigasi.



Gambar 5.7. Pandu Zahri

Saudara kembar dari Zahra Mandu, bertugas memandu juga tapi dalam sesi interogasi.



Gambar 5.8. Susan Pectora

Tersangka utama. Apakah dia benar-benar membunuh pada umur yang muda?

Map



Gambar 5.9. *Airport*

Tempat *spawn* pertama. Akan ada pengenalan singkat *main character*, Aziz Tan, dan penjelasan tentang kasus.



Gambar 5.10. *Bazaar*

Bazaar yang diadakan dekat dengan TKP. Penyelenggara memaksa untuk menyelenggarakannya walau ada investigasi polisi. Beberapa bukti akan ditemukan di sini bersama dengan saksi pada nantinya.



Gambar 5.11. *Jalan*

Tempat terjadinya pembunuhan ganda. *CCTV* hanya menangkap bagian pembunuhan pertama, beberapa saat setelah itu pembunuhan kedua terjadi.



Gambar 5.12. Kantor Polisi

Setelah sesi investigasi selesai *player* akan pergi ke sini untuk interogasi orang-orang yang bersangkutan dengan kasus.

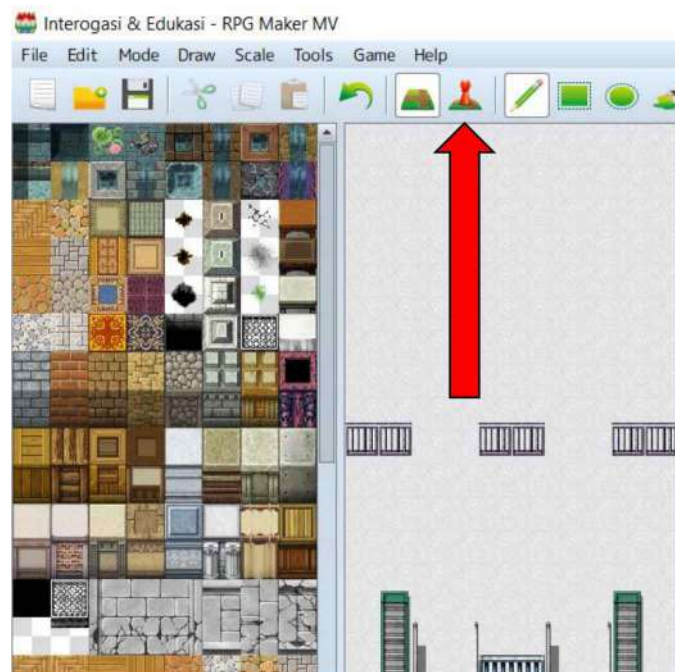


Gambar 5.13. Ruang Interogasi

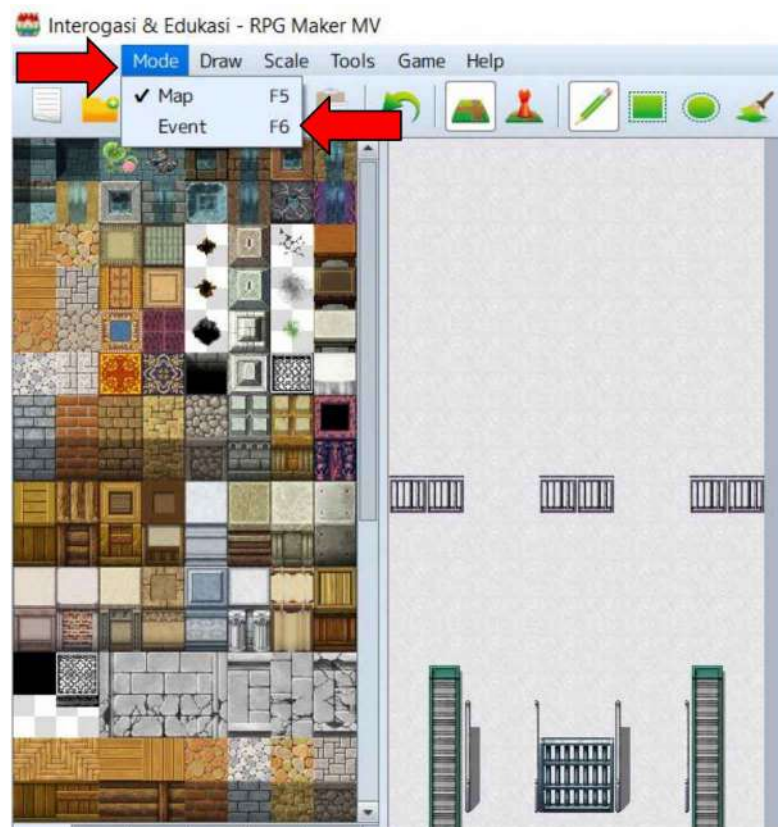
Tempat menginterogasi orang-orang yang berhubungan dengan kasus. *Player* bisa menanyakan lebih lanjut bagian testimoni *character* agar mendapatkan info lebih lanjut. *Player* harus menunjukkan bukti yang benar untuk *progress*.

Tahap Pembuatan

1. Setelah selesai membuat *map* dan *character*, Anda dapat langsung membuat *event* untuk game Anda dengan cara klik ikon *event* atau bisa menggunakan *menu Mode* di bagian atas atau menggunakan *hotkey* F6.

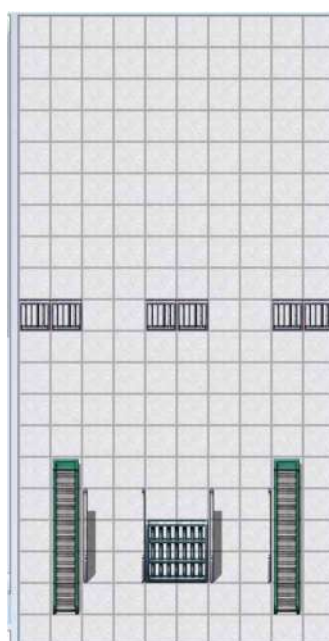


Gambar 5.14. Ikon *event* pada RPG Maker MV



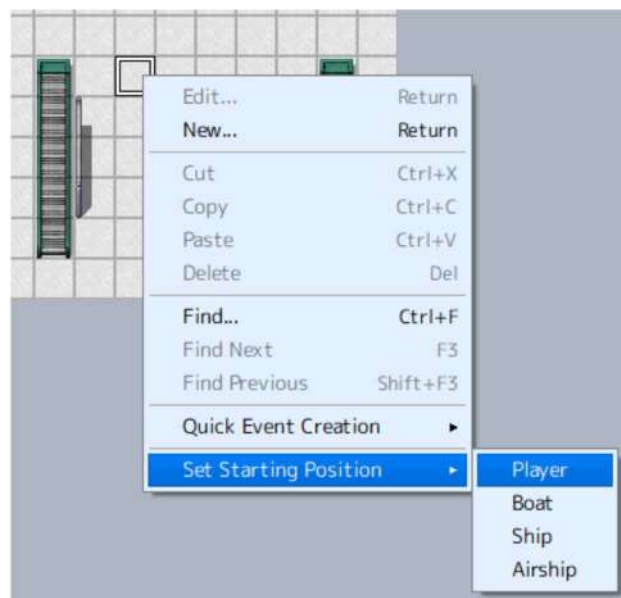
Gambar 5.15. *Menu Mode* dan *hotkey* untuk *event* pada RPG Maker MV

2. Bila *event* sudah aktif, maka akan muncul kotak-kotak pada *map* seperti ini.



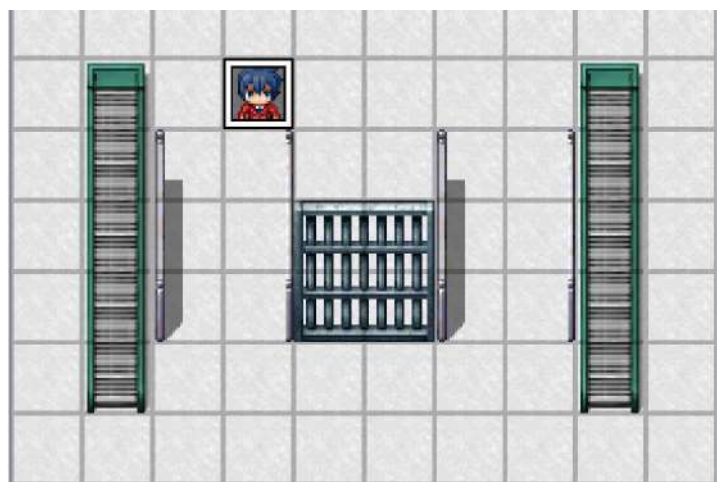
Gambar 5.16. RPG Maker MV pada *mode event*

3. Berikutnya adalah membuat *cutscene* untuk *map* pertama, tetapi sebelum membuat *cutscene* kita harus memberi player starting position dengan cara klik kanan di salah satu kotak dan pilih *Set Starting Position* lalu pilih *Player*.



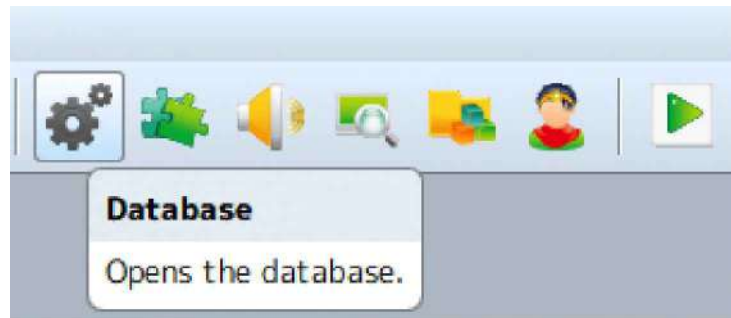
Gambar 5.17. Pemilihan posisi awal *player*

4. Selanjutnya akan muncul *character* yang Anda buat.



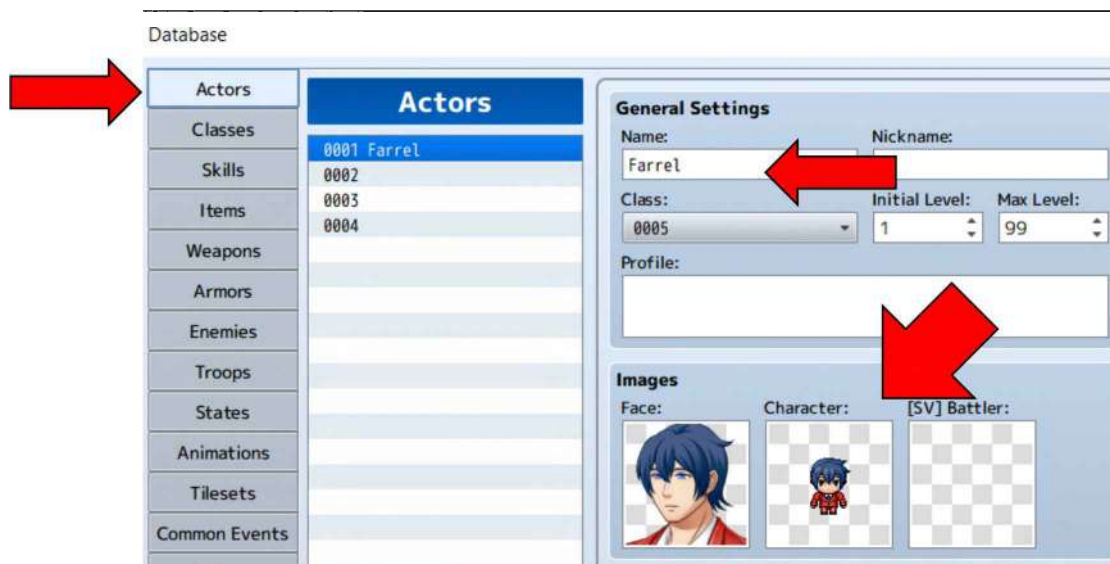
Gambar 5.18. Posisi awal Player

5. Bila *character* tidak muncul, cobalah pergi ke *Database*.



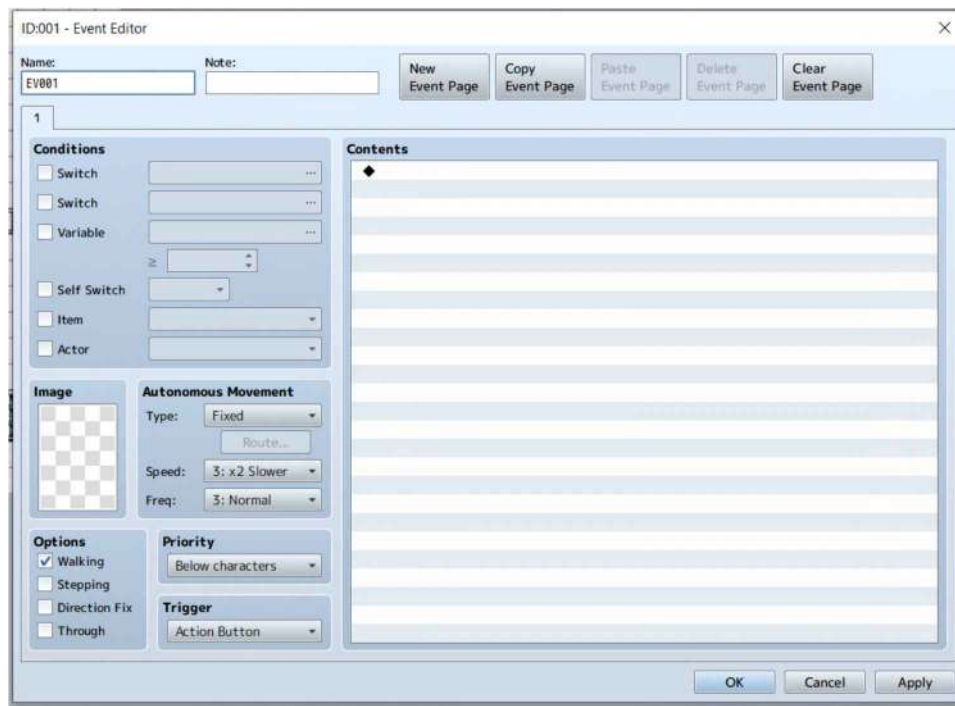
Gambar 5.19. Ikon *Database* pada RPG Maker MV

6. Lalu, masuk ke bagian *Actors* dan pastikan *character* Anda ada di situ. Jangan lupa untuk *Apply* setelah menambahkan sesuatu di RPG Maker.



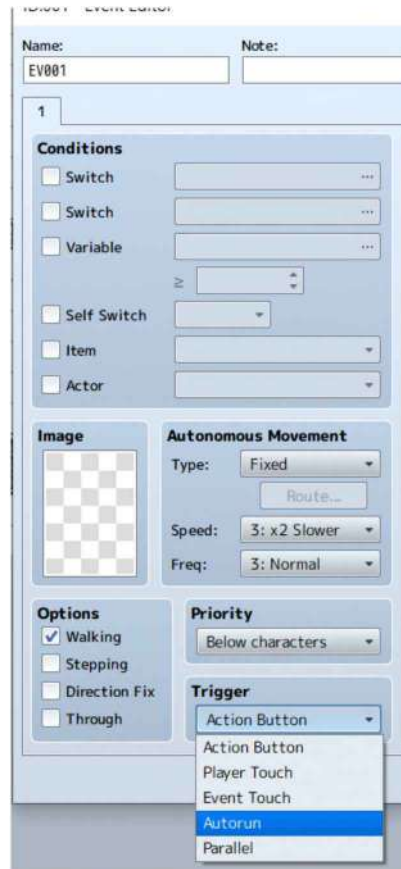
Gambar 5.20. Halaman *Database Actors*

7. Untuk pembuatan cutscene, Anda bisa dua kali klik pada salah satu kotak dan akan muncul *Event Editor*.



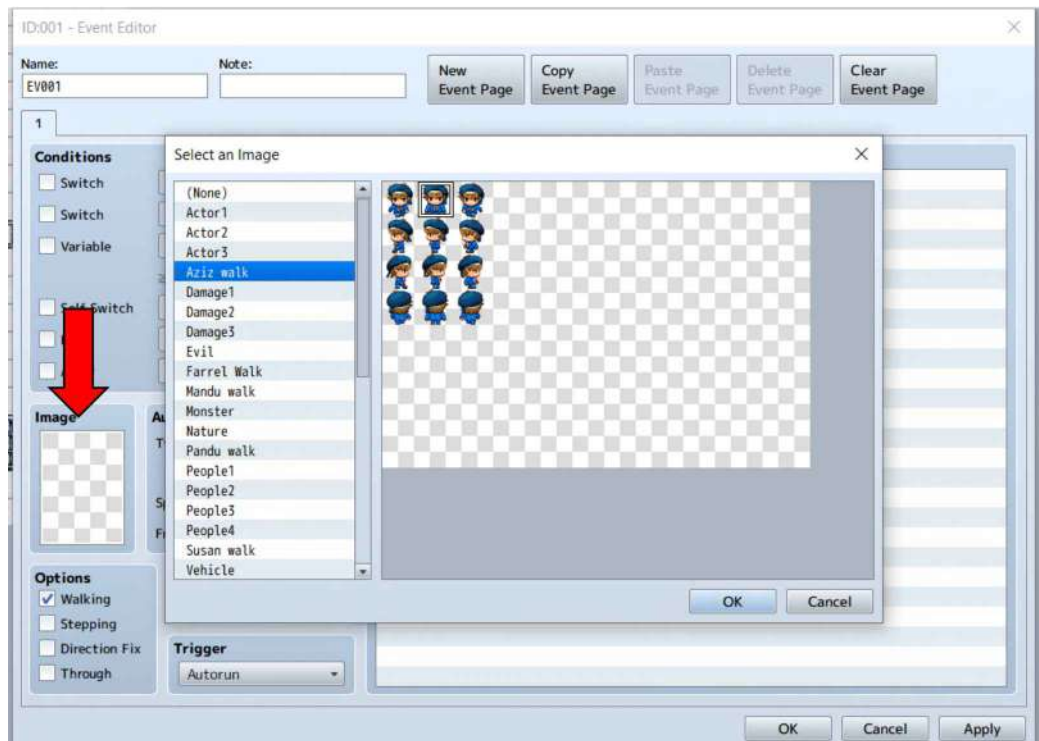
Gambar 5.21. Halaman *Event Editor*

8. Karena cutscene yang akan dibuat langsung jalan saat *player start game* maka *Trigger* diganti menjadi *Autorun*.



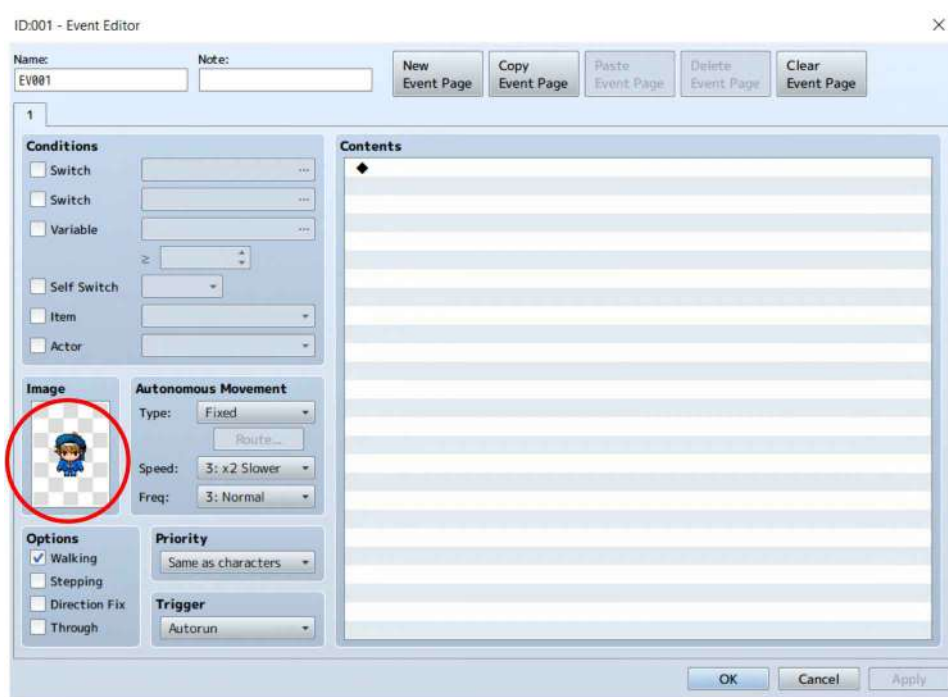
Gambar 5.22. Penggantian *Trigger*

9. Jika Anda akan membuat *cutscene* menggunakan *character player* saja Anda bisa langsung mem-*program*, tetapi bila ingin ada interaksi antara 2 orang maka Anda bisa memberi gambar *character* kedua di bagian *Image*.



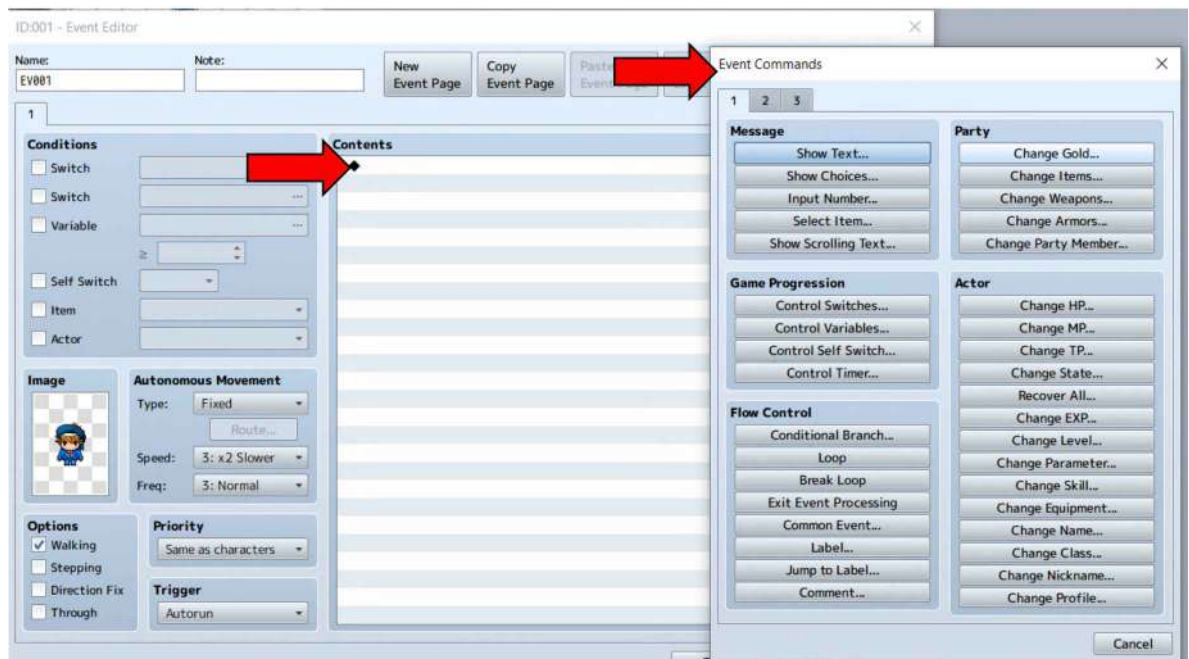
Gambar 5.23. Halaman *Image* pada *Event Editor*

10. Pilihlah salah satu *sprite character* dan klik *OK*, maka *character* Anda akan muncul di *Image*.



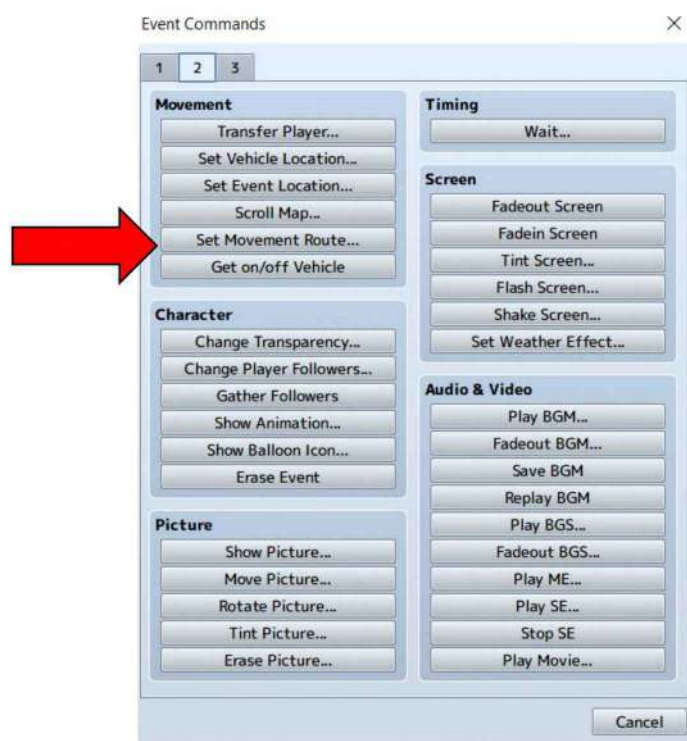
Gambar 5.24. Gambar *character* dalam *Image*

11. Sekarang untuk program, klik dua kali pada bagian *Contents*. Akan muncul halaman *Event Commands*.



Gambar 5.25. Halaman *Event Commands* pada *Event Editor*

12. Setelah halaman *Event Commands* keluar pilihlah *Set Movements Route* pada halaman 2 untuk menggerakkan *character*.



Gambar 5.26. Halaman 2 *Event Commands*

13. Klik *Set Movement Route* untuk menjalankan *character player*, pilih *Player* pada pojok kiri atas. Sedangkan, untuk menjalankan *character kedua*, ganti *Player* menjadi *This Event*.



Gambar 5.27. Halaman *Set Movements Route*

14. *Program pergerakan character sesuai keinginan. Setiap program yang ditambahkan merupakan 1 kotak.*



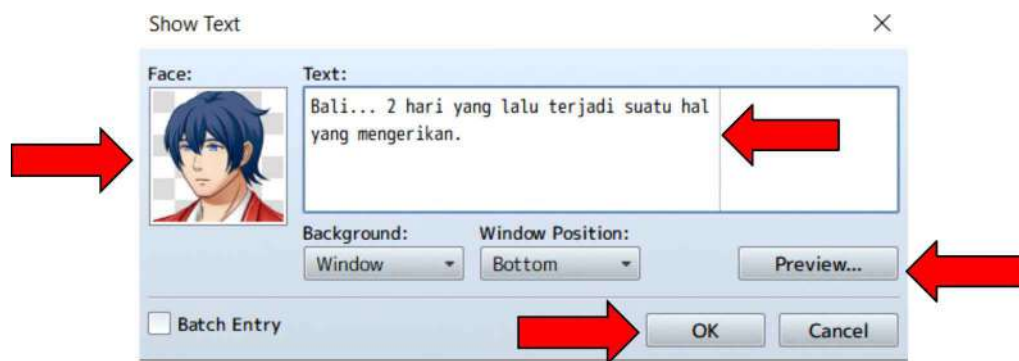
Gambar 5.28. *Program pergerakan character*

15. Untuk menambahkan percakapan klik *Show Text* pada halaman 1 *Event Commands*.

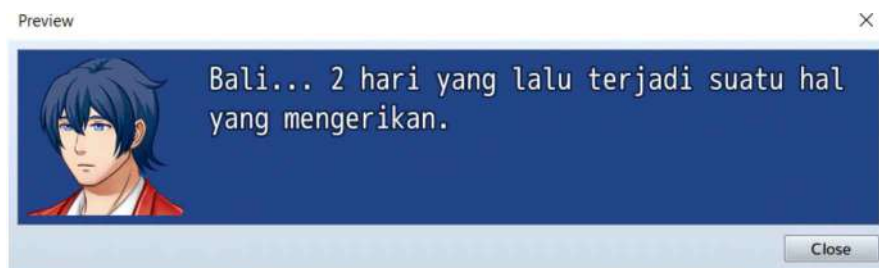


Gambar 5.29. Halaman 1 *Event Commands*

16. Tulis *text* yang ingin ditulis pada kotak *Text*, bila ingin memberi gambar *character* saat *text* muncul masukkan salah satu muka *character* di kotak *Face*. Setelah selesai Anda bisa klik *Preview* untuk melihat hasil jadinya, lalu klik *OK* untuk konfirmasi.

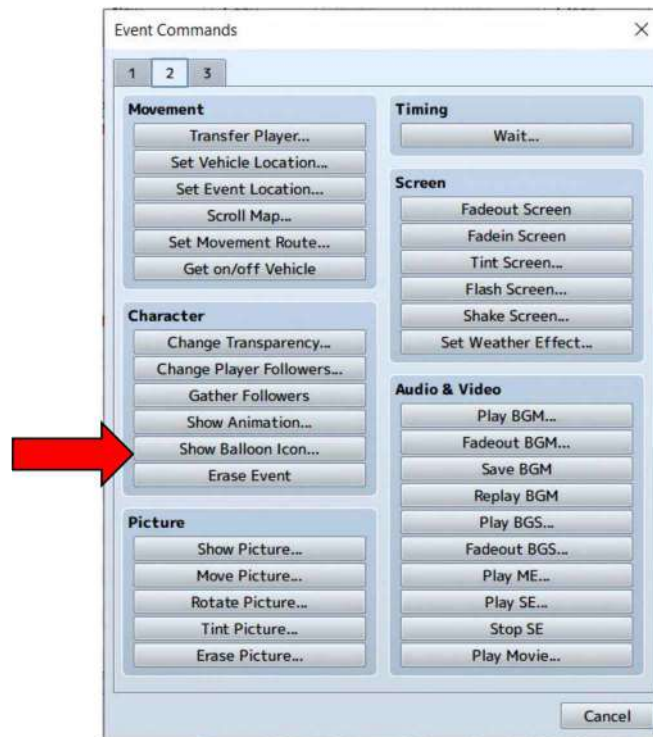


Gambar 5.30. Halaman *Show Text*



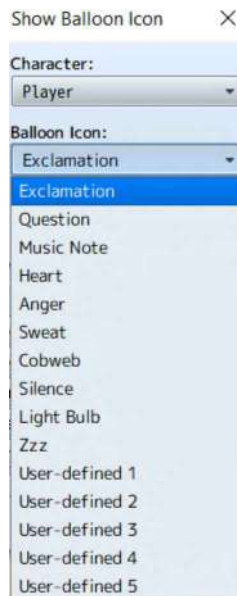
Gambar 5.31. *Preview Text*

17. Selain *text* di RPG Maker MV Anda bisa menambahkan ikon *balloon* untuk membuat game lebih jelas dan menarik. Mungkin terlihat seperti hal kecil, tetapi terkadang hal-hal inilah yang menambah nilai dalam game. Ikon *balloon* bisa diakses lewat halaman 2 *Event Commands* dengan klik *Show Balloon Icon*.



Gambar 5.32. Halaman 2 *Event Commands*

18. Seperti biasa Anda bisa mengganti apakah ikon akan berada di *character player* atau *event*, dan tentu saja Anda juga bisa mengubah ikon yang muncul. Selain kedua hal tersebut, juga ada *checkbox* untuk *Wait for Completion*. *Checkbox* ini digunakan tergantung kondisi, misalkan Anda ingin ikon *balloon* terlihat bersamaan dengan *text* atau aksi yang *character* lakukan, maka tidak perlu diberi centang. Bila, ingin ikon *balloon* terlihat dulu sebelum *text* atau aksi setelahnya tampak, maka beri centang.



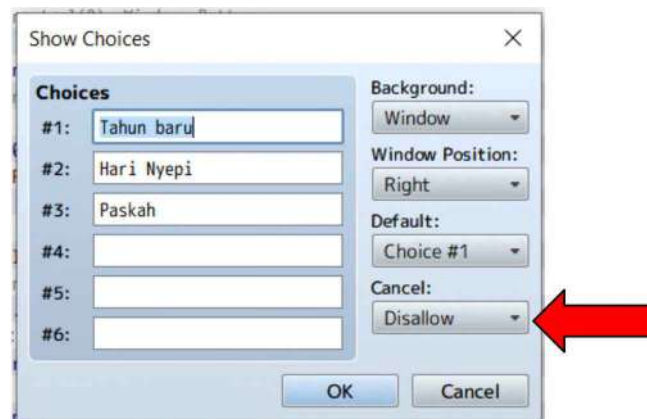
Gambar 5.33. Halaman *Show Balloon Icon*

19. Untuk memberikan pilihan pada *player* gunakanlah *Show Choices* pada halaman 1 *Event Commands*.



Gambar 5.34. Halaman 1 *Event Commands*

20. Karena pilihan yang akan diberikan tidak memiliki pilihan untuk membatalkan maka gantilah *Cancel* menjadi *Disallow*. Setelah itu, Anda bisa mengisi jenis-jenis pilihan di *Choices*. Bila sudah selesai dan sudah klik *OK* maka Anda bisa memberikan *program* tambahan di bawah setiap pilihan.



Gambar 5.35. Halaman *Show Choices*

21. Berikutnya Anda bisa memberi *commands* lainnya di bawah setiap pilihan untuk memberi hasil dari pilihan yang ada.

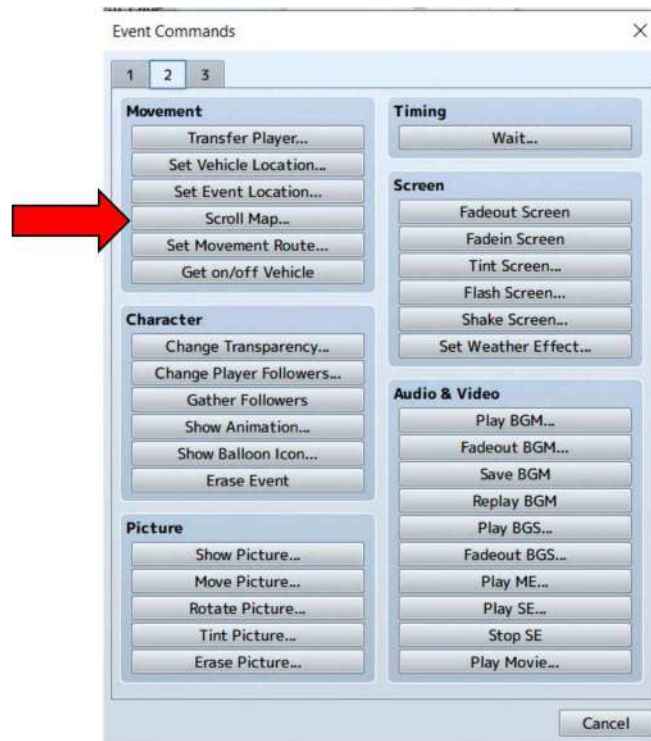
```

: When Tahun baru
◆Text : Farrel what(0), Window, Bottom
:      : (Hm?! Apa yang kupikirkan barusan?)
◆
: When Hari Nyepi
◆Text : Farrel neutral(0), Window, Bottom
:      : (Benar juga, Hari Nyepi kalau begitu
:      : bukannya berarti pada hari itu tidak
:      : ada..?)
◆Show Choices : Korban, Saksi (Window, Right, #1, -)
: When Korban
◆Text : Farrel ack(0), Window, Bottom
:      : ?! (Kalau tidak ada korban kenapa aku di
:      : sini?!)
◆
: When Saksi
◆Text : Farrel what(0), Window, Bottom
:      : (Benar juga... kalau hari itu adalah Hari
:      : Nyepi pasti tidak banyak orang keluar dari
:      : rumah.)
◆
: End
◆
: When Paskah
◆Text : Farrel what(0), Window, Bottom
:      : (Eh? Itu masih bulan depan.)

```

Gambar 5.36. *Program pilihan*

22. Lalu, jika Anda ingin menunjukkan sesuatu tanpa *character player* bergerak maka Anda bisa menggunakan *Scroll Map* pada halaman 2 *Event Commands*. *Command* ini akan menggerakkan kamera saja. Bila menggunakan *command* ini pastikan ada *command* lagi di bawah *Scroll Map* karena jika tidak ada, *Scroll Map* tidak akan jalan.

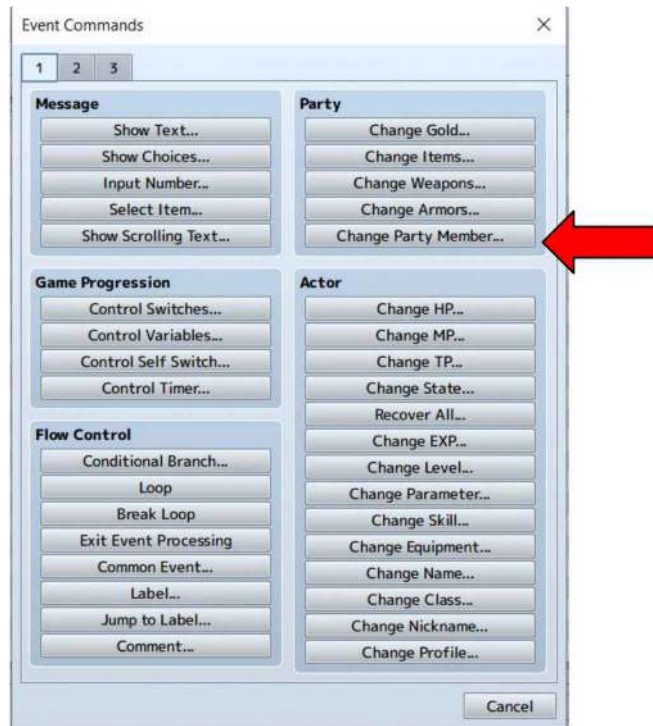


Gambar 5.37. Halaman 2 *Event Commands*

```
◆ Scroll Map : Up, 5, 4
◆ Text : Farrel smile(0), Window, Bottom
      : Oh, hey Aziz.
```

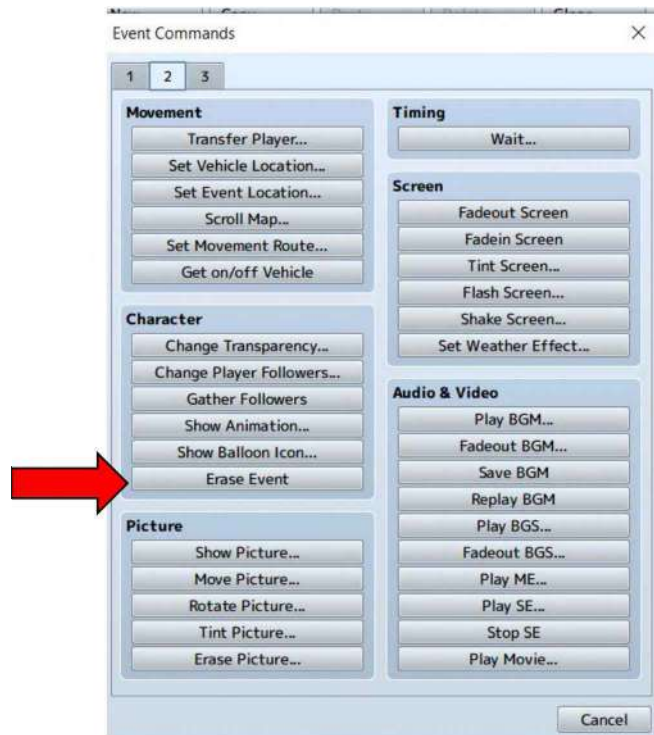
Gambar 5.38. *Program Scroll Map*

23. Sekarang karena *character player* akan berpindah tempat, Anda bisa gunakan *Transfer Player* pada halaman 2 *Event Commands*, tetapi sebelum itu karena dalam interaksi tadi terdapat 2 *character* maka harus *Change Party Member* terlebih dahulu yang terletak pada halaman 1 *Event Commands* agar dalam visual masuk akal, bila Anda tidak menambahkan *character* kedua maka hanya *player* saja yang terlihat akan pergi.



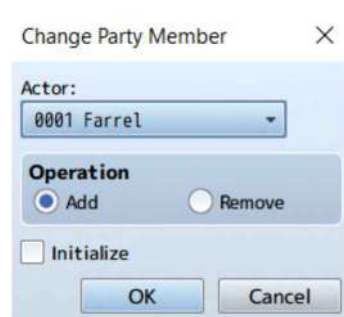
Gambar 5.39. Halaman 1 *Event Commands*

24. Sebelum *Change Party Member* supaya *character* kedua pada *Image event* hilang kita harus menggunakan *Erase Event* pada halaman 2 *Event Commands*. Ini dilakukan agar tidak ada *double character* di *map* dan satunya di *party*.



Gambar 5.40. Halaman 2 *Event Commands*

25. Untuk menggunakan *Change Party Member* yang perlu Anda lakukan hanyalah memilih *character* mana yang ingin ditambahkan atau dikurangi dari *party* sekarang. Jangan lupa untuk menambahkan *character* yang ingin ditambahkan atau dikurangi di *Database*.



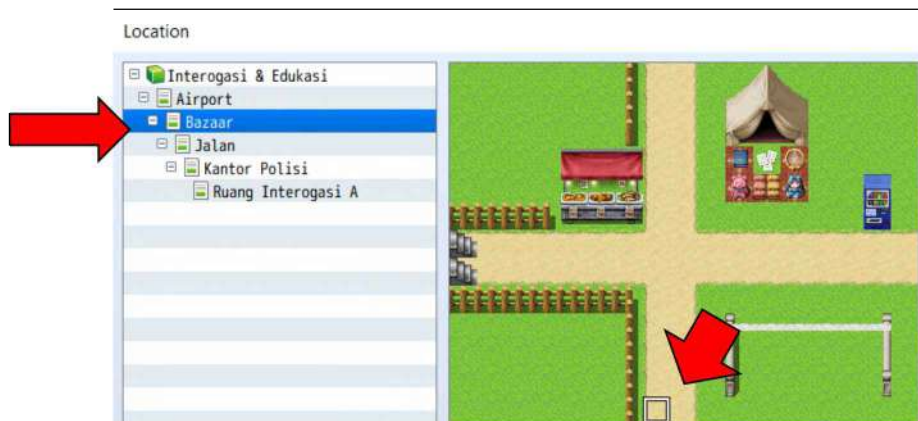
Gambar 5.41. Halaman *Change Party Member*

26. Lalu, untuk memindahkan *player* dengan *Transfer Player* Anda akan diminta untuk memilih lokasi dimana Anda ingin memindah *player*. Caranya, klik satu kali pada *Direct*

designation. Setelah itu, pilih map dan kotak mana yang Anda inginkan untuk *player spawn*.

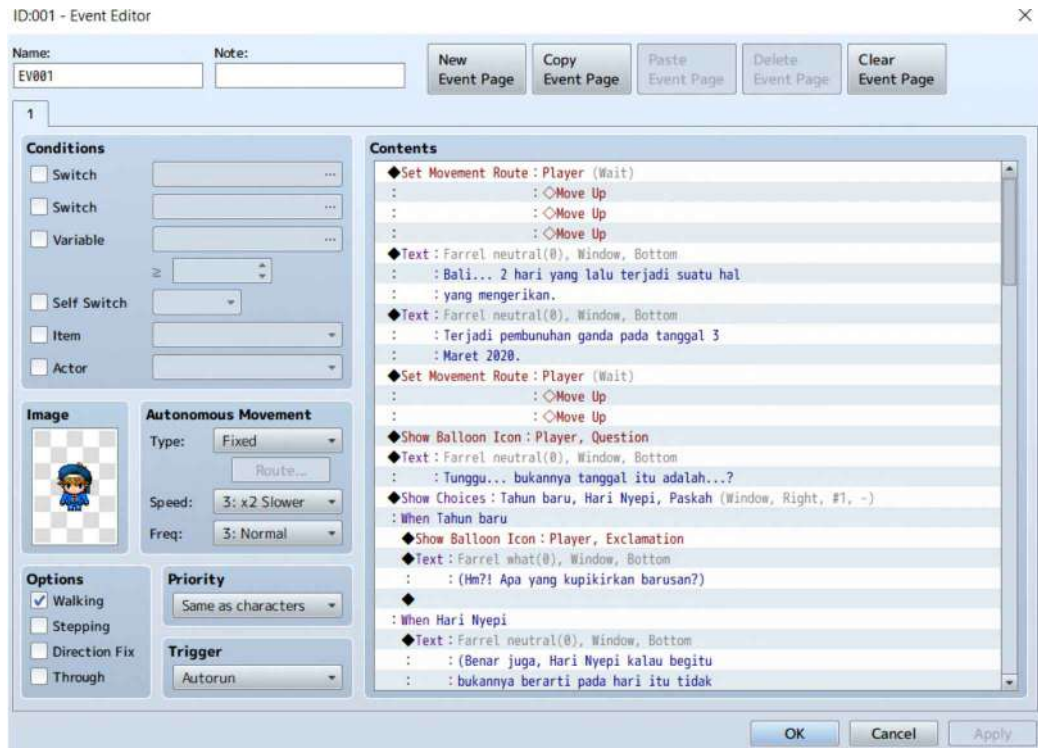


Gambar 5.42. Halaman *Transfer Player*

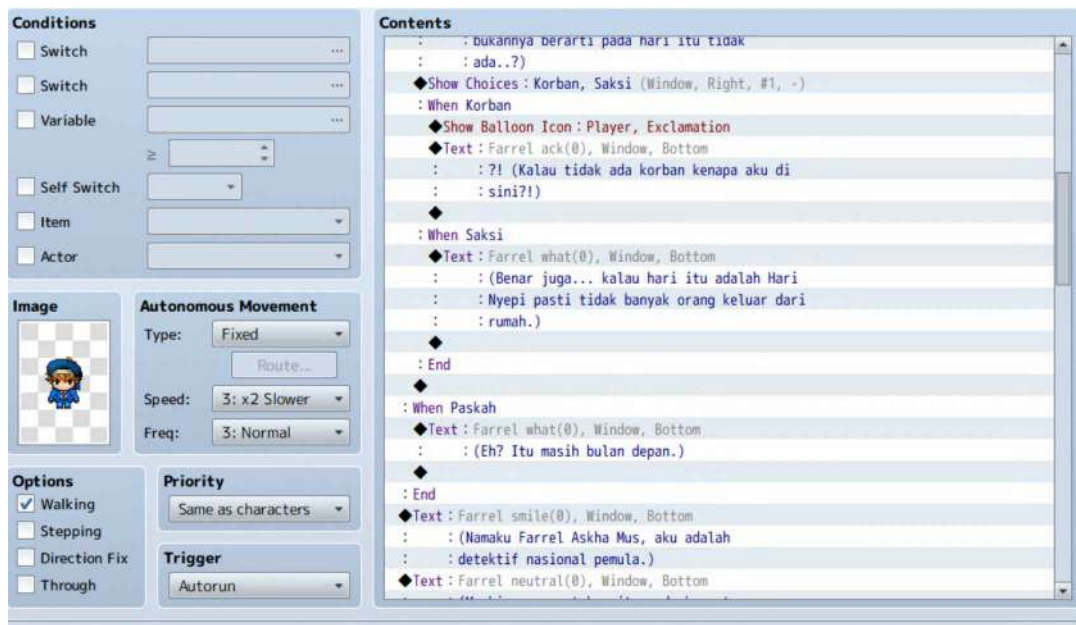


Gambar 5.43. Halaman *Location*

27. Berikut hasil *program* untuk *map Airport*



Gambar 5.44. Program cutscene Airport bagian 1



Gambar 5.45. Program cutscene Airport bagian 2



Gambar 5.46. Program cutscene Airport bagian 3

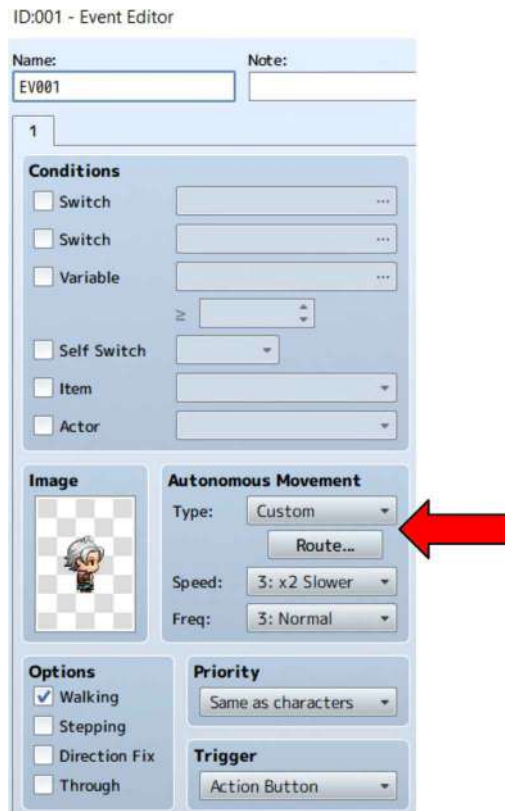


Gambar 5.47. Program cutscene Airport bagian 4



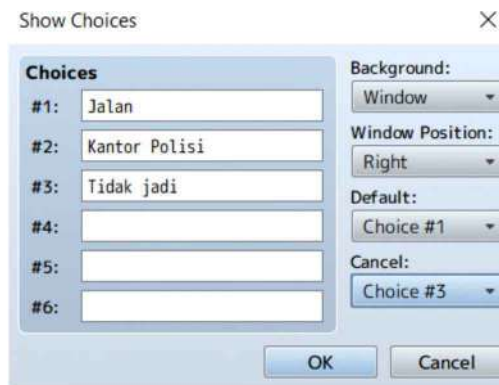
Gambar 5.48. Program cutscene Airport bagian 5

28. Untuk *map* kedua yaitu Bazaar, akan memiliki *character* yang berjalan dengan sendirinya tetapi bukan *cutscene*. Jadi, *character* ini akan jalan terus menerus dalam pola tertentu tanpa mengganggu pergerakan *player*. Sama seperti membuat *cutscene*, Anda perlu memberi *Image* tetapi, alih-alih mengisi bagian *Contents* pada *Event Editor*, Anda mengisi bagian *Autonomous Movement*. Untuk pergerakan sesuai rute yang ditentukan pilihlah *Custom* pada *Type* lalu, klik *Route* dan pilih rute.

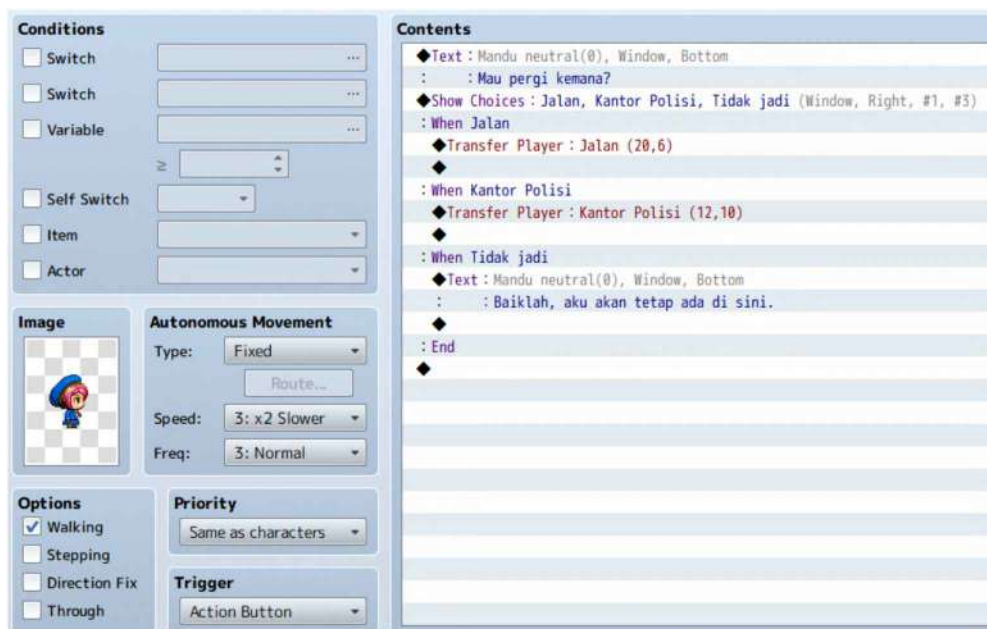


Gambar 5.49. Halaman *Event Editor*

29. Dari *map Bazaar* dan seterusnya *player* akan berpindah tempat dengan interaksi pada salah satu *character*. Untuk melakukan ini maka gunakanlah *command Show Choices*. Lalu, gunakan *Transfer Player* di bawah pilihan yang ada, karena pilihan kali ini merupakan pilihan yang bisa dibatalkan maka bagian *Cancel* diganti menjadi pilihan mana yang digunakan untuk membatalkan interaksi. Juga, pakailah *Action Button* dan bukan *Autorun* untuk *Trigger* agar *event* tidak jalan dengan sendirinya dan *loop*.



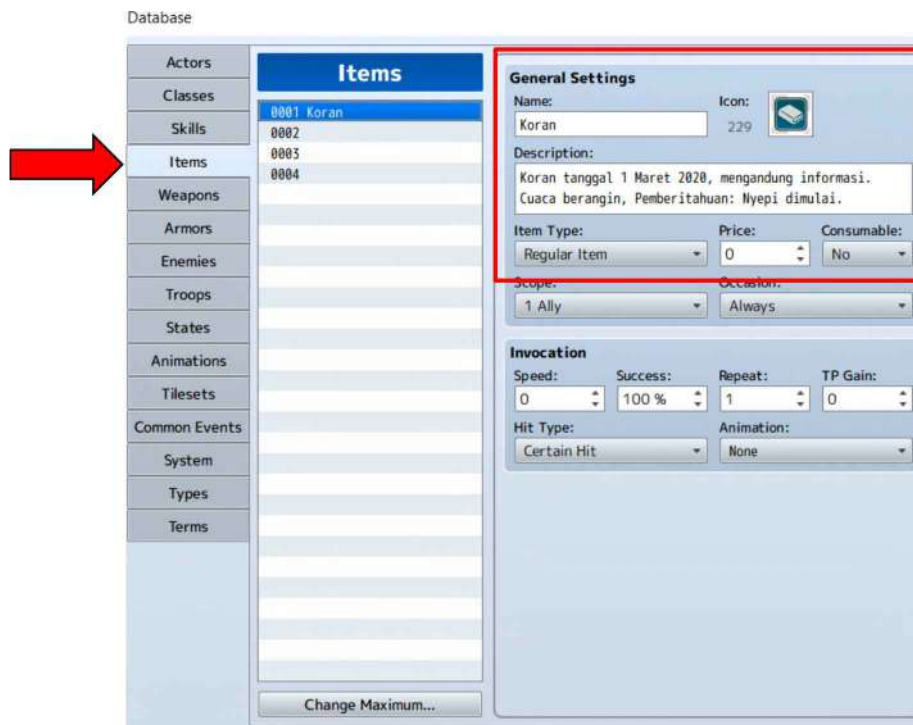
Gambar 5.50. *Program Show Choices perpindahan player*



Gambar 5.51. *Program perpindahan player*

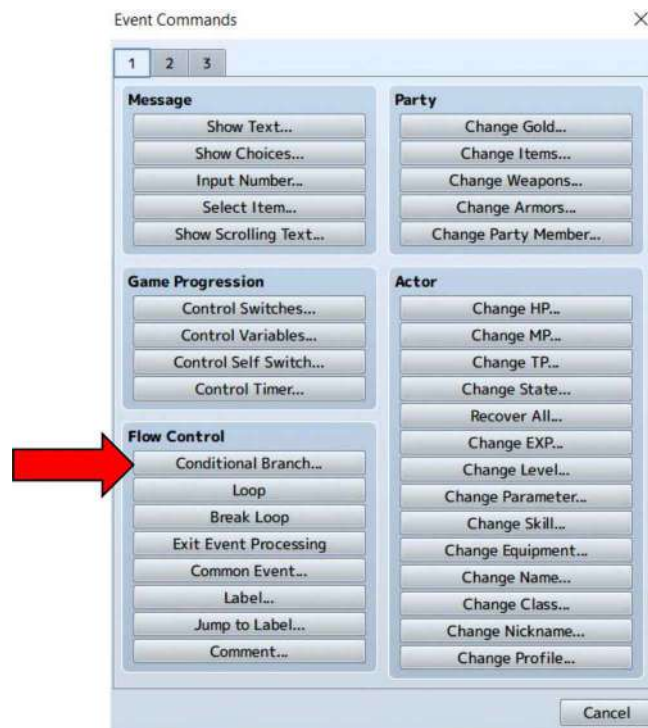
30. Sekarang *player* sudah memasuki sesi investigasi, maka diperlukan barang atau bukti yang berhubungan dengan kasus. Pergi ke *Database* dan klik *Items*. Di sana Anda bisa mengganti nama barang, ikon barang, deskripsi barang, dan lain-lain. Game ini akan mengganti bagian nama, ikon, deskripsi, dan *Consumables* (apakah barang bisa

digunakan atau tidak) saja. Bila ingin menambah jumlah barang maka bisa gunakan *Change Maximum*.



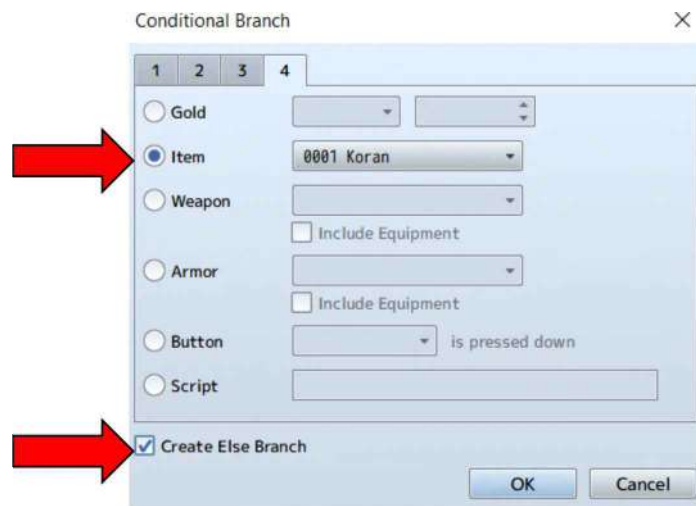
Gambar 5.52. Halaman *Database Items*

31. Setelah itu, ada *event* yang memiliki *Conditional Branch* pada halaman 1 *Event Commands*.



Gambar 5.53. Halaman 1 *Event Commands*

32. *Conditional Branch* bekerja saat *player* memiliki suatu barang, *event* lain sudah terjadi, dan lain-lain. Bisa dibayangkan kerjanya seperti pilihan tetapi apa yang ada dalam game yang memilih, bukan *player*. Game ini akan menggunakan *Item*. Klik *Conditional Branch* dan pergi ke halaman 4, pilih *Item*, pilih barang yang akan digunakan dan centang *Create Else Branch*.



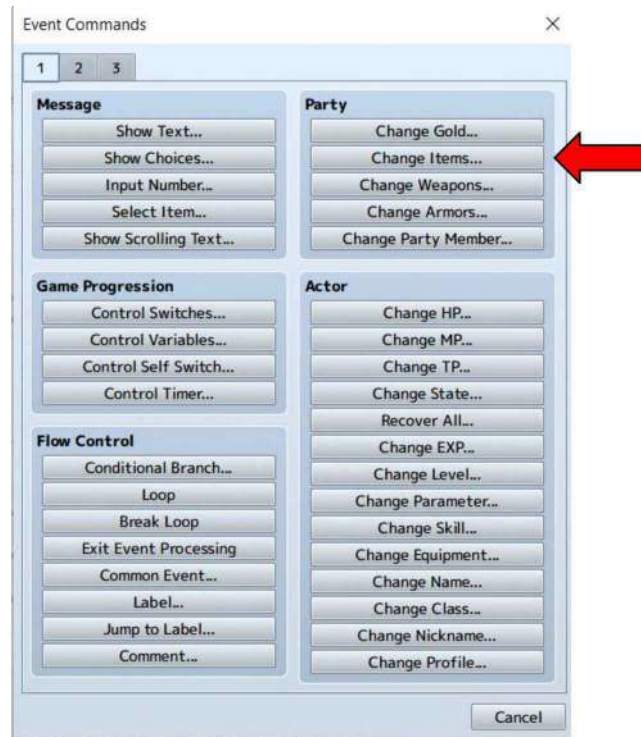
Gambar 5.54. Halaman 4 *Conditional Branch*

33. *Program* dibawah *If* berarti bila *party* memiliki koran sedangkan *Else* saat *party* tidak memiliki koran. Sama seperti pilihan, Anda bisa menambahkan *command* di bawahnya untuk menjadi kelanjutan dari *If* dan *Else*.



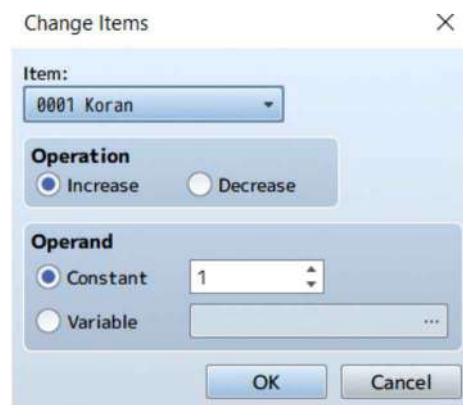
Gambar 5.55. *Program If* dan *Else*

34. Karena *player* akan diberi sebuah barang pada *program Else* maka bisa menggunakan *Change Items* pada halaman 1 *Event Commands*.



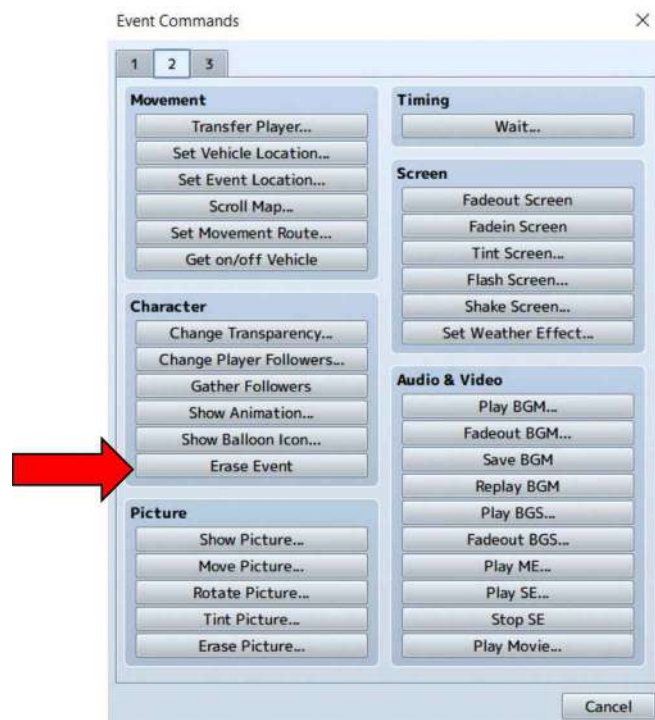
Gambar 5.56. Halaman 1 *Event Commands*

35. Di *Change Items* Anda bisa menambahkan atau mengurangi barang *player* serta menentukan jumlah barang yang akan didapatkan atau hilang, dan tentunya barang apa.



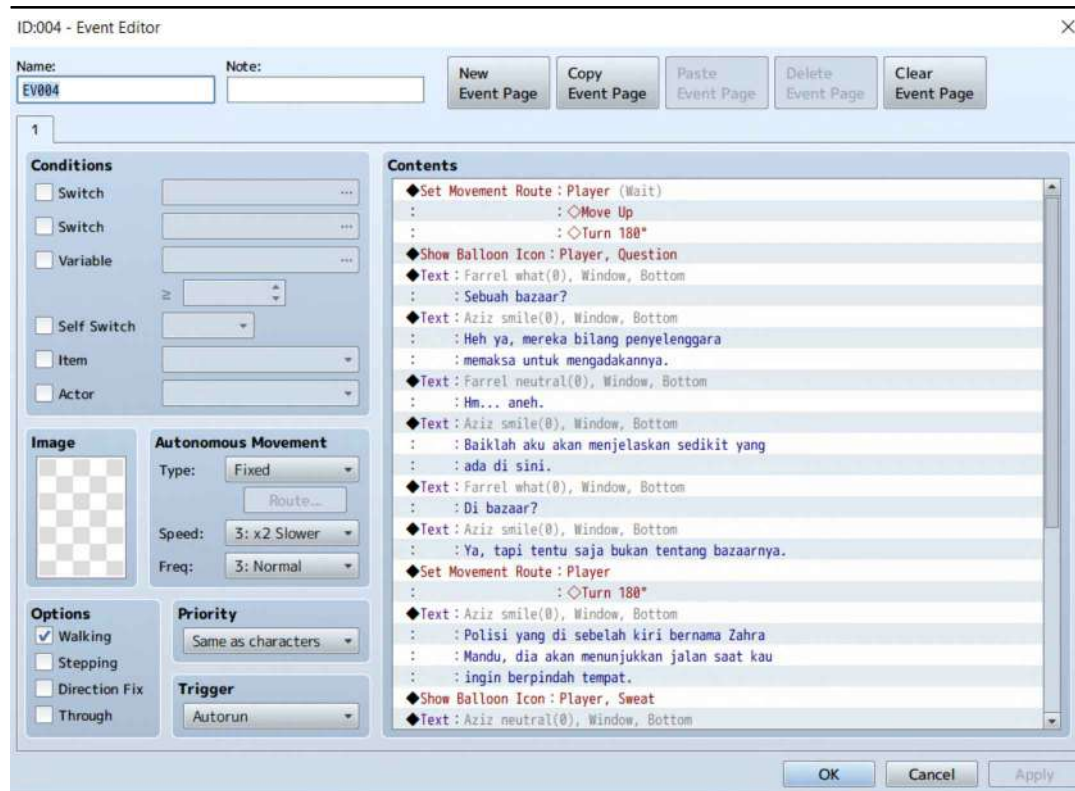
Gambar 5.57. Halaman *Change Items*

36. Saat membuat *cutscene* pada map yang akan *player* bisa jelajah maka setelah selesai harus diberi *Erase Event* agar *cutscene* tidak terulang.

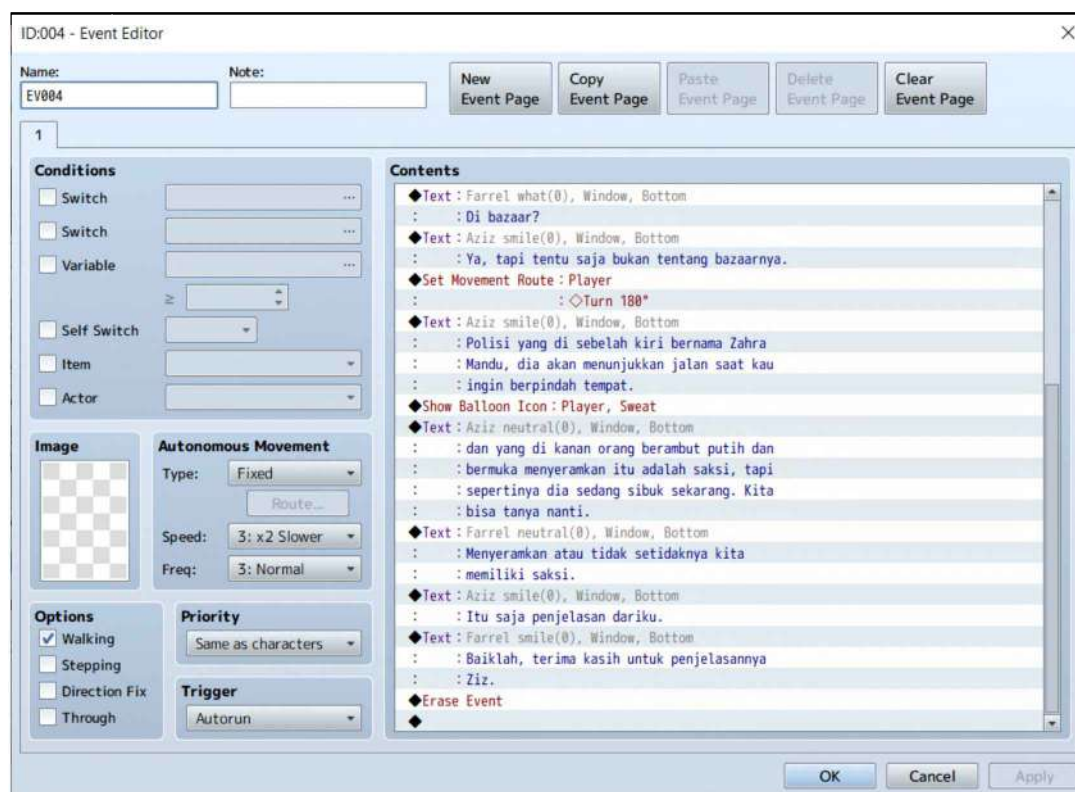


Gambar 5.58. Halaman 2 *Event Commands*

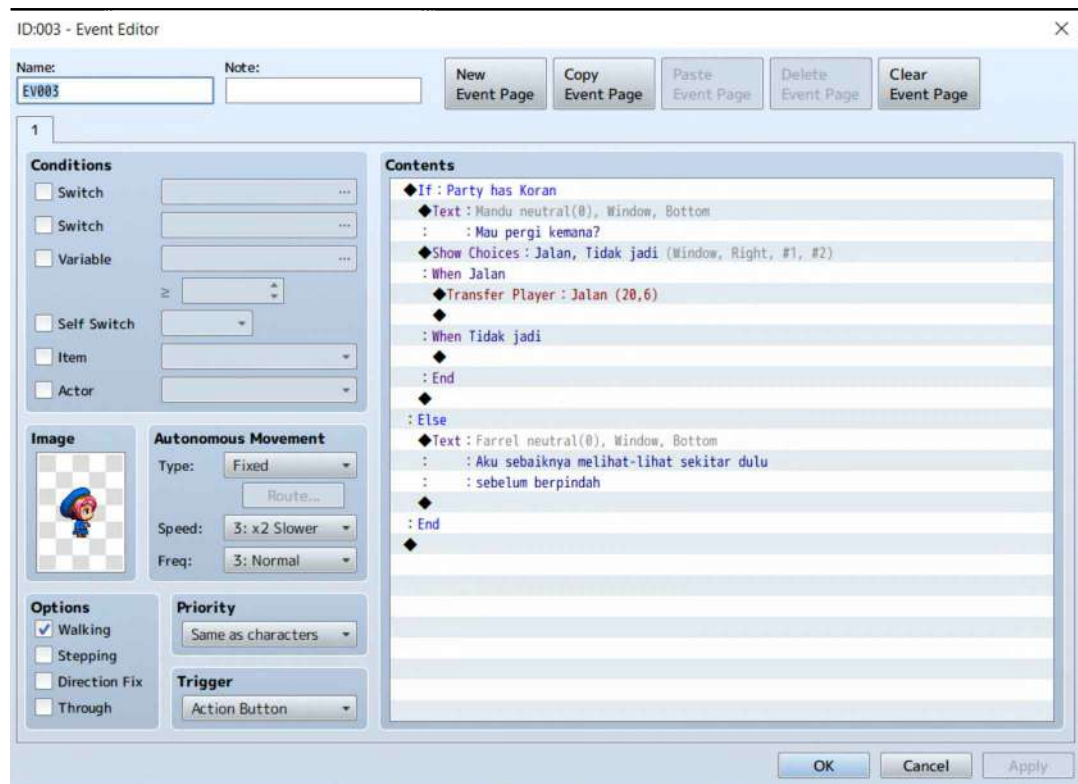
37. Berikut hasil *program* untuk *map Bazaar*



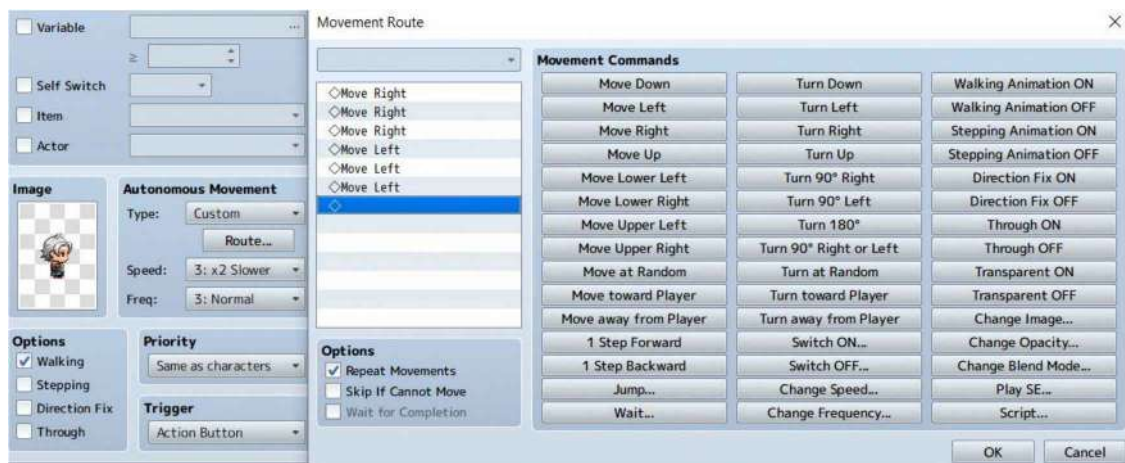
Gambar 5.59. Program cutscene bagian 1



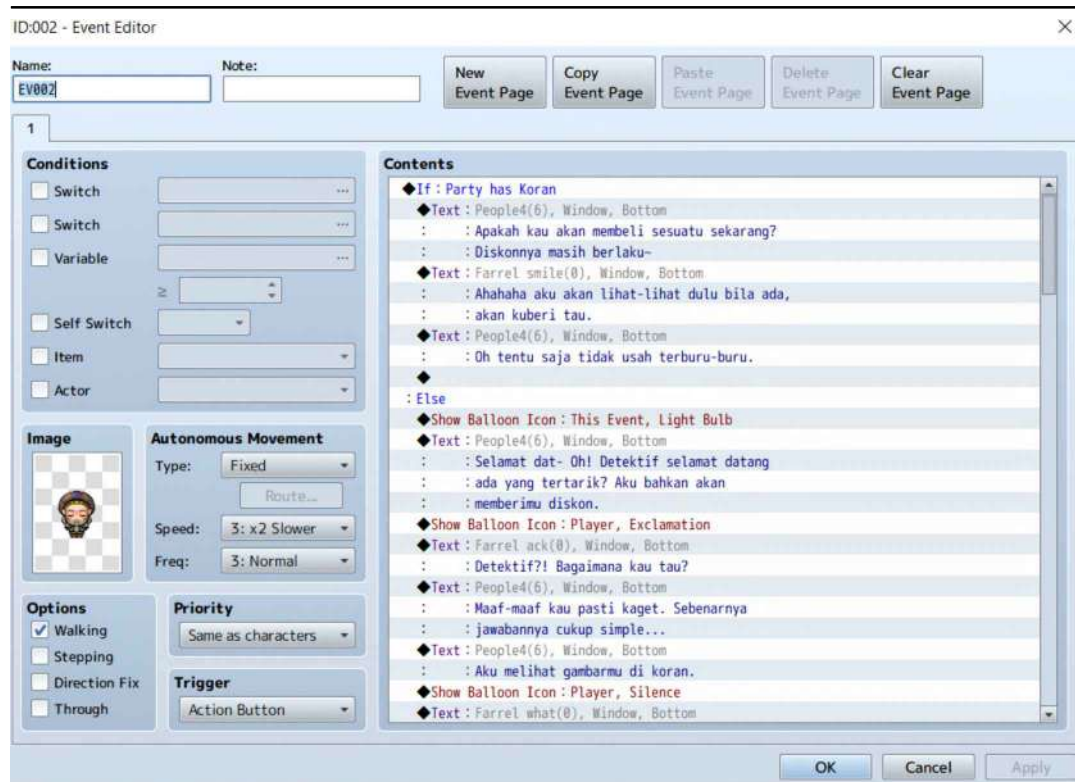
Gambar 5.60. Program cutscene bagian 2



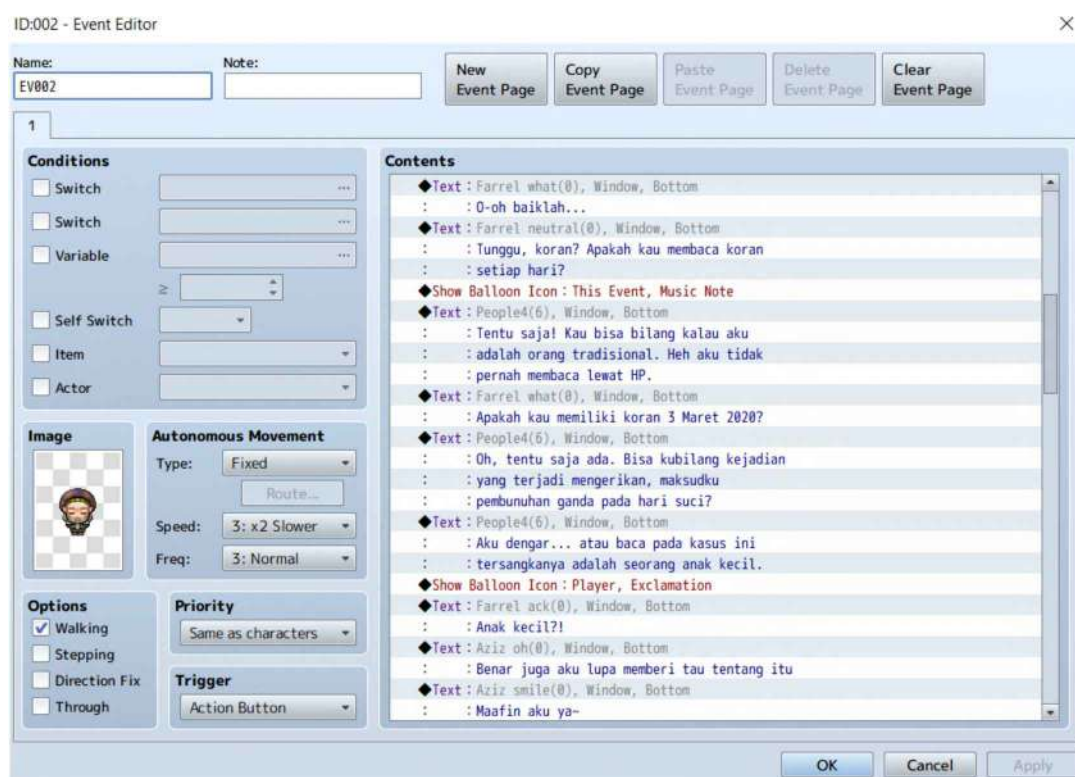
Gambar 5.61. Program perpindahan *player*



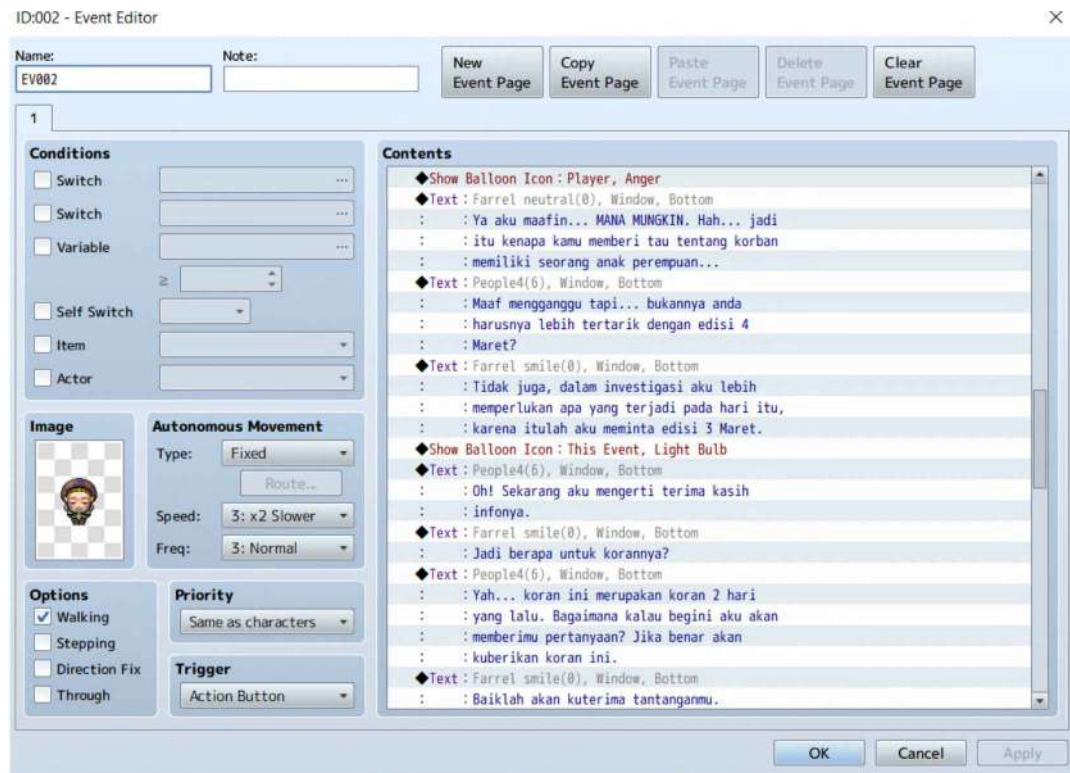
Gambar 5.62. Program pergerakan mengulang *NPC*



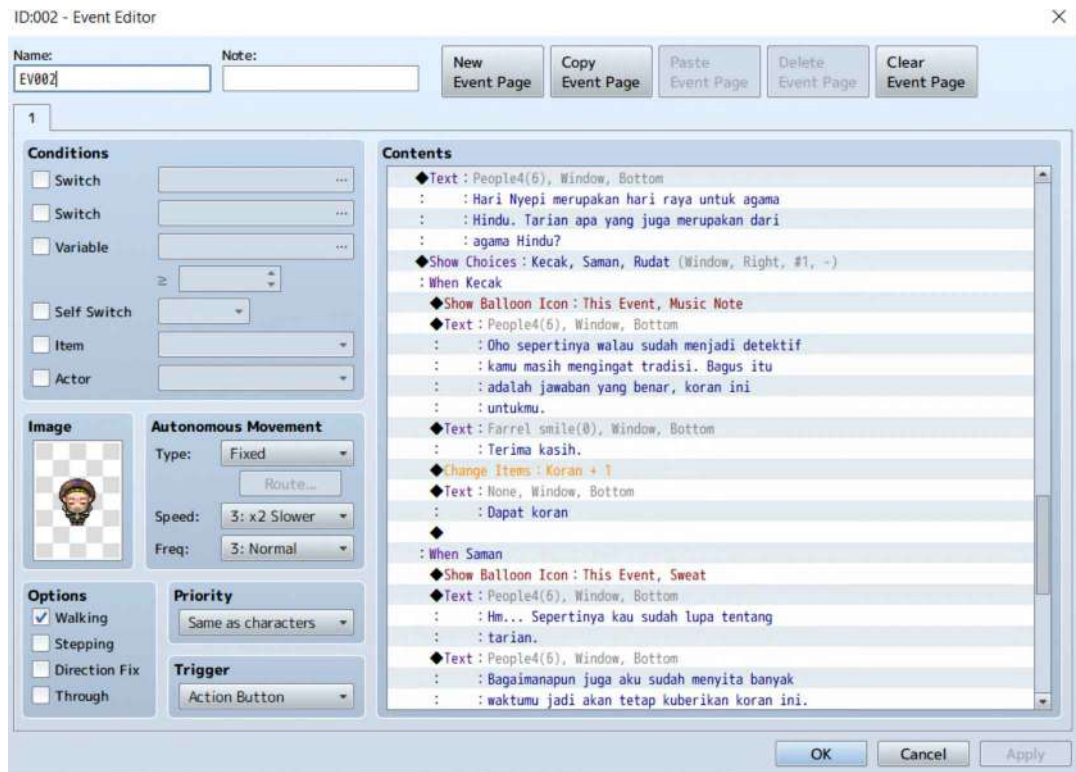
Gambar 5.63. Program *If, Else*, dan *Change Items* bagian 1



Gambar 5.64. *Program If, Else, dan Change Items* bagian 2

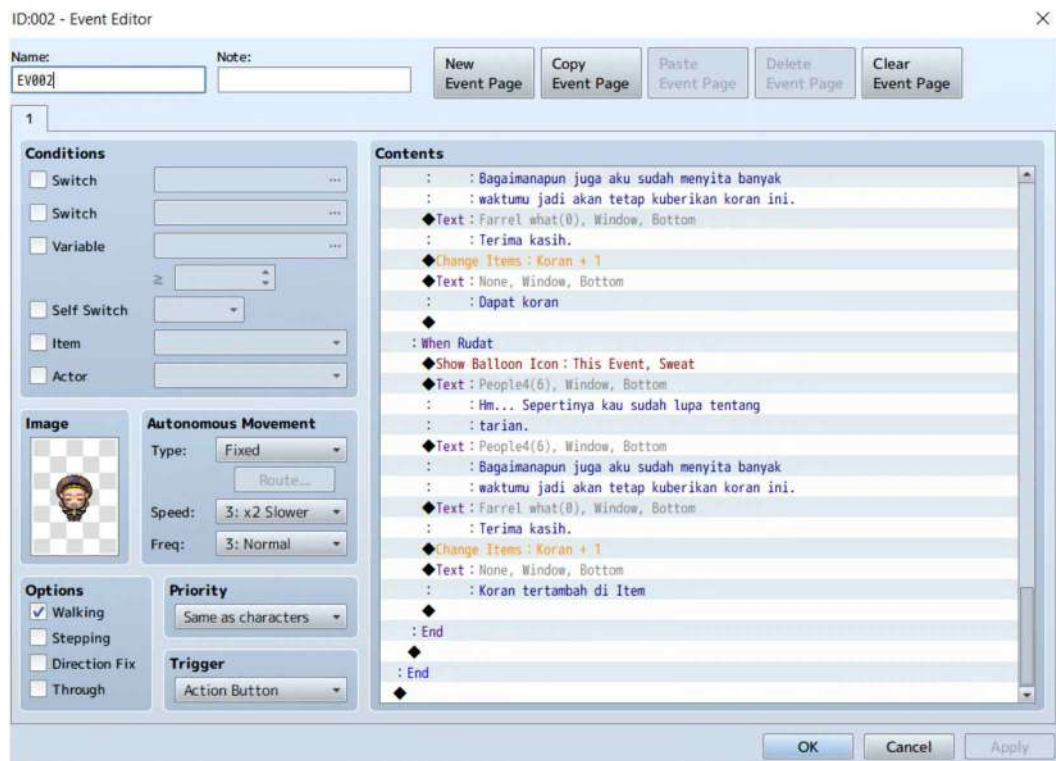


Gambar 5.65. *Program If, Else, dan Change Items* bagian 3



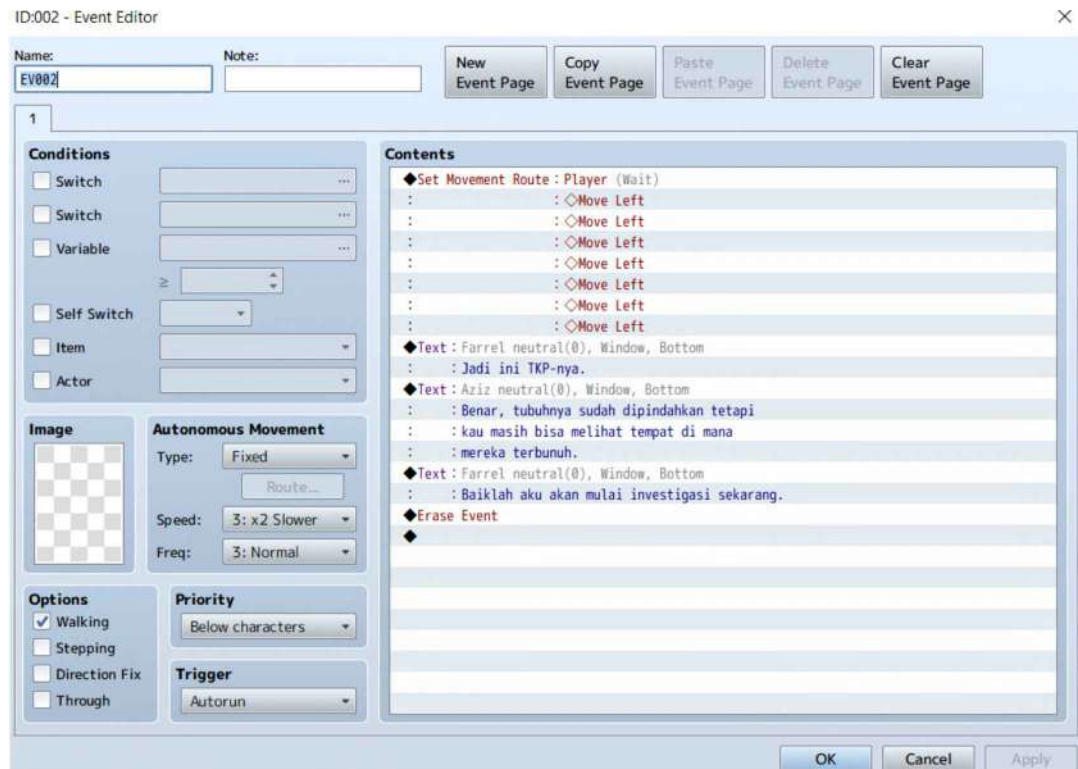
Gambar 5.66. Program If, Else, dan Change Items bagian

4

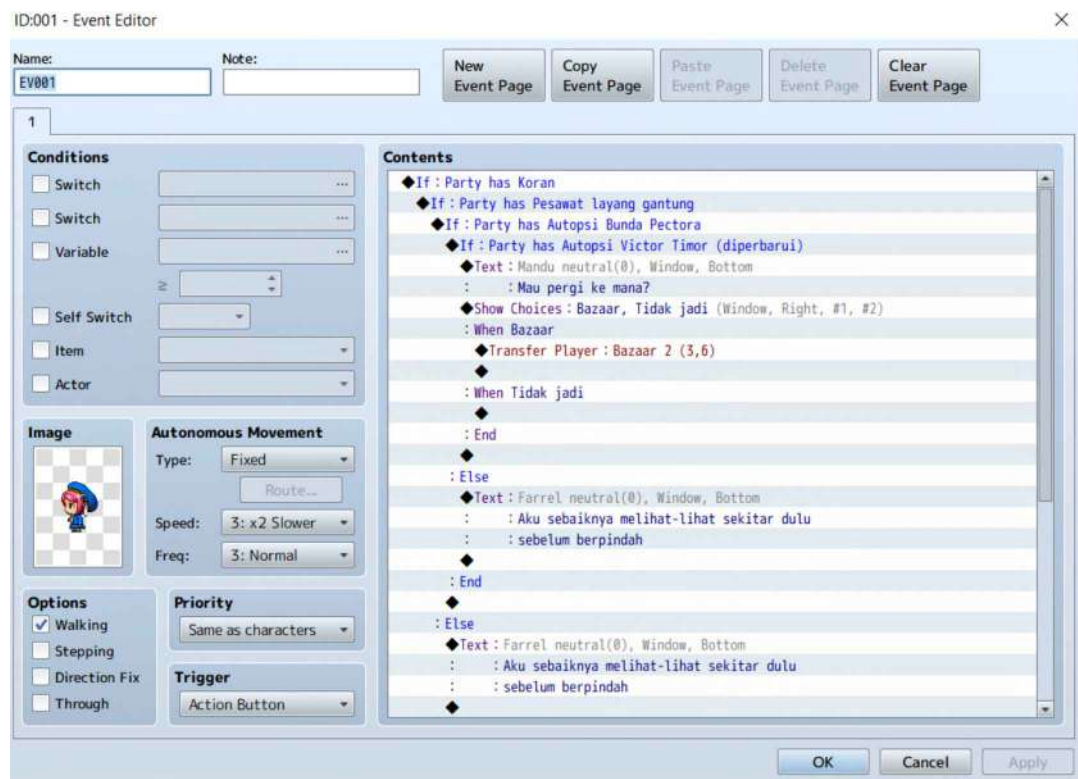


Gambar 5.67. *Program If, Else, dan Change Items* bagian
5

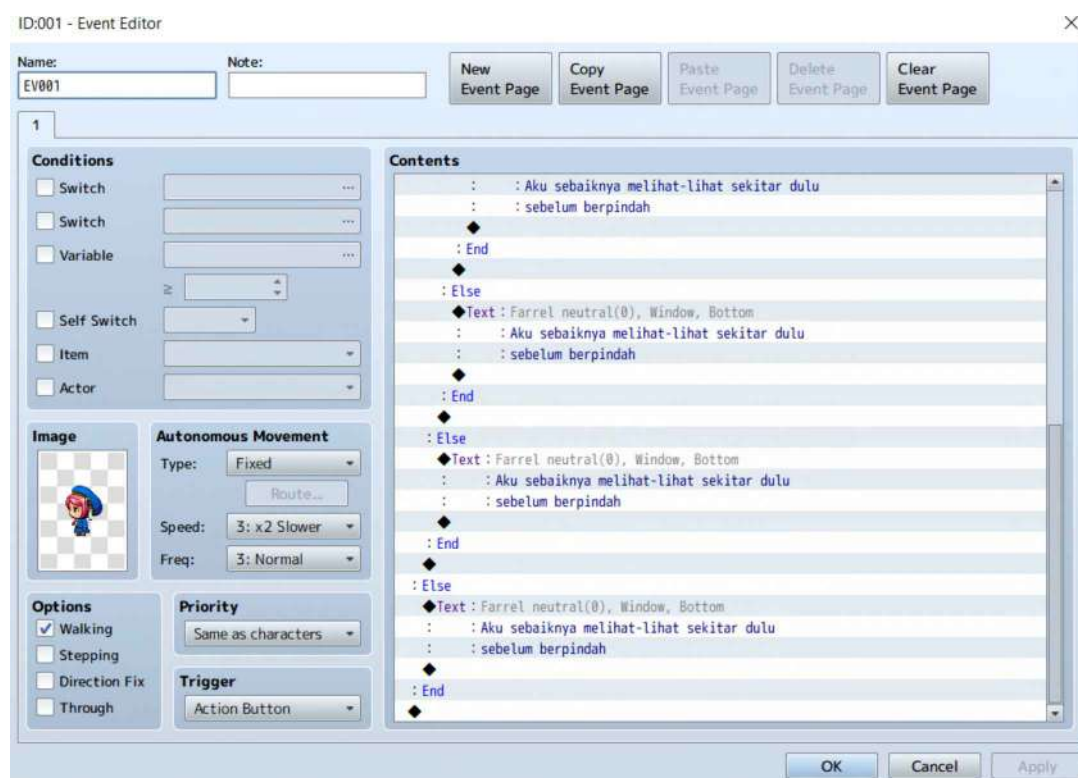
38. Berikut hasil *program* untuk *map* Jalan



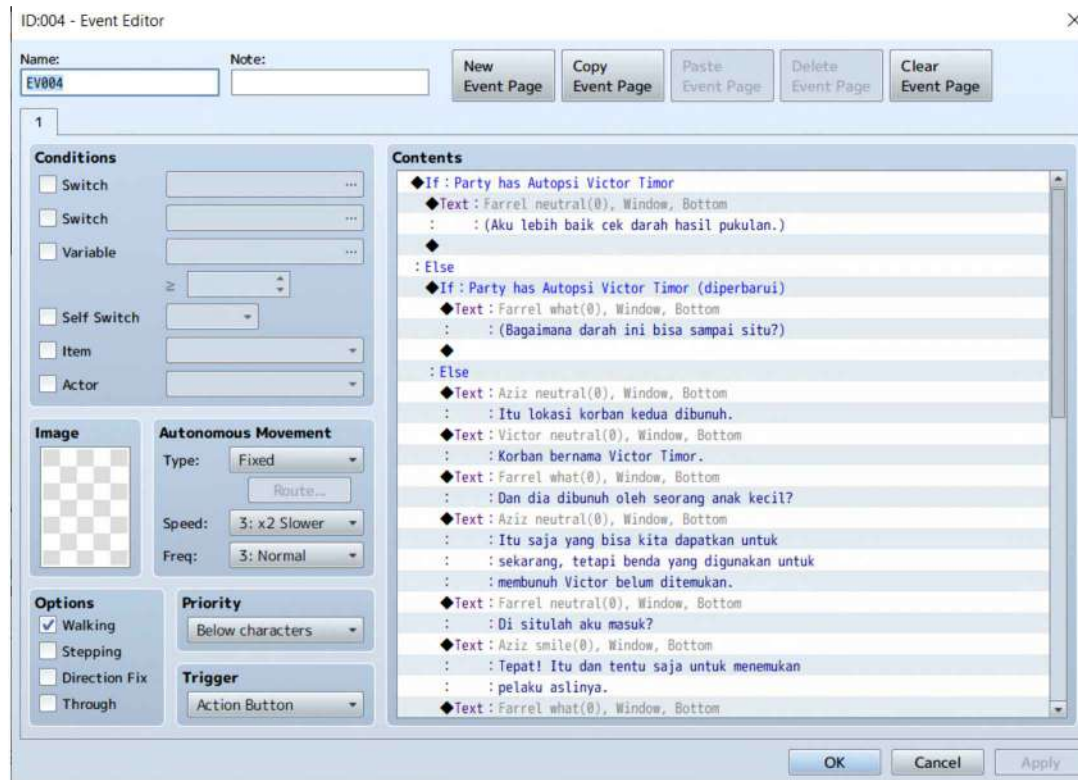
Gambar 5.68. *Program cutscene*



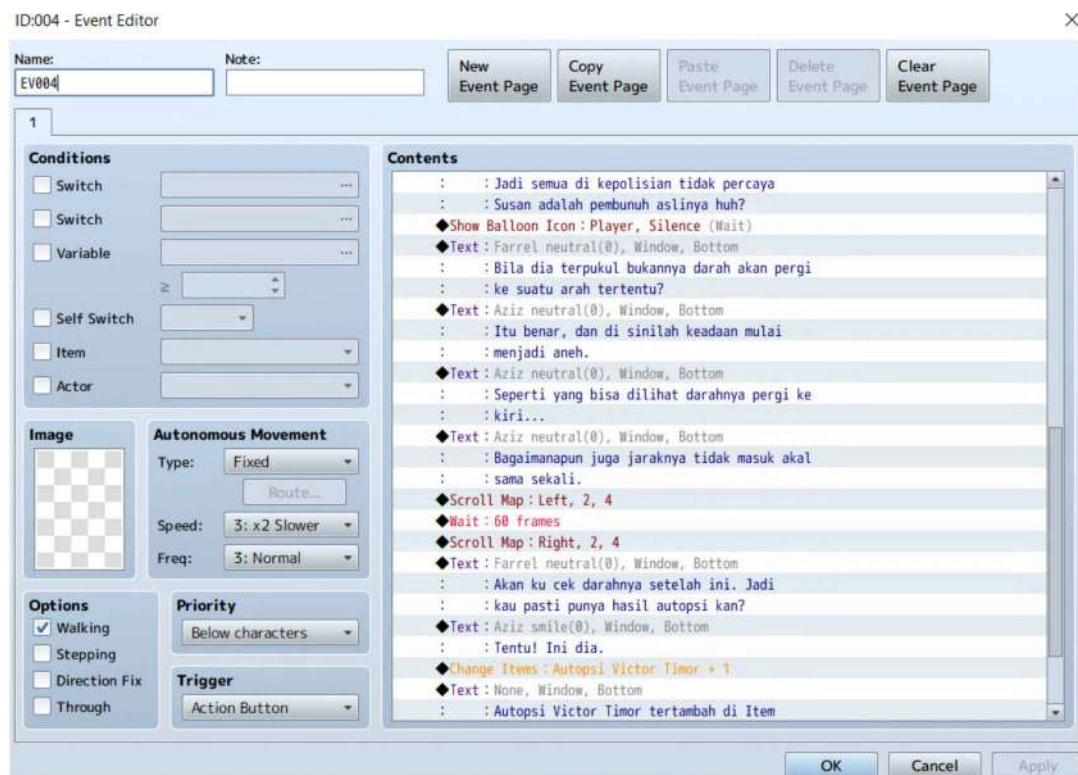
Gambar 5.69. Program perpindahan *player* bagian 1



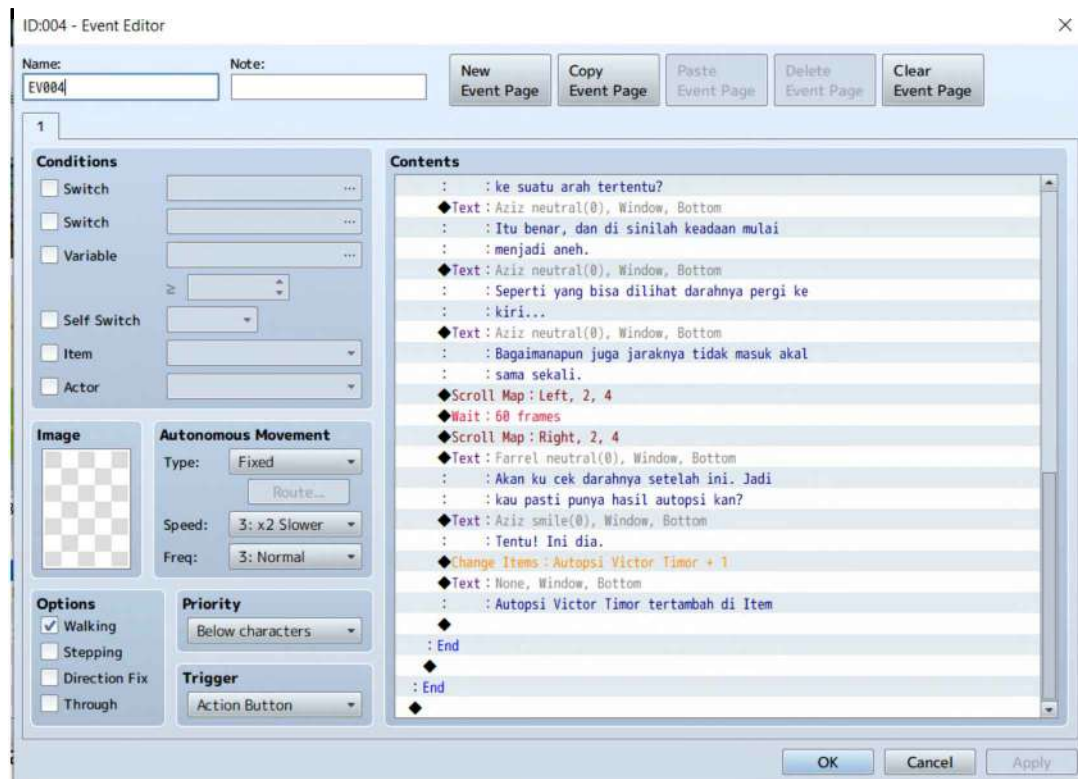
Gambar 5.70. Program perpindahan *player* bagian 2



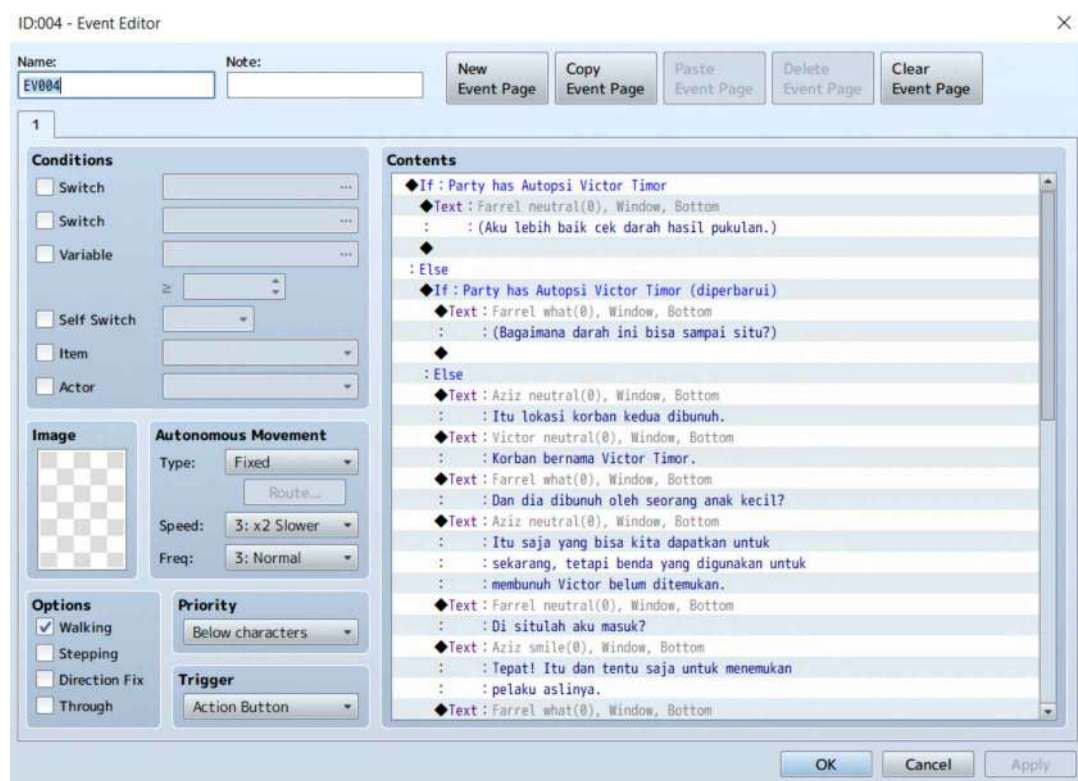
Gambar 5.71. Program TKP (Tempat Kejadian Perkara) Bunda bagian 1



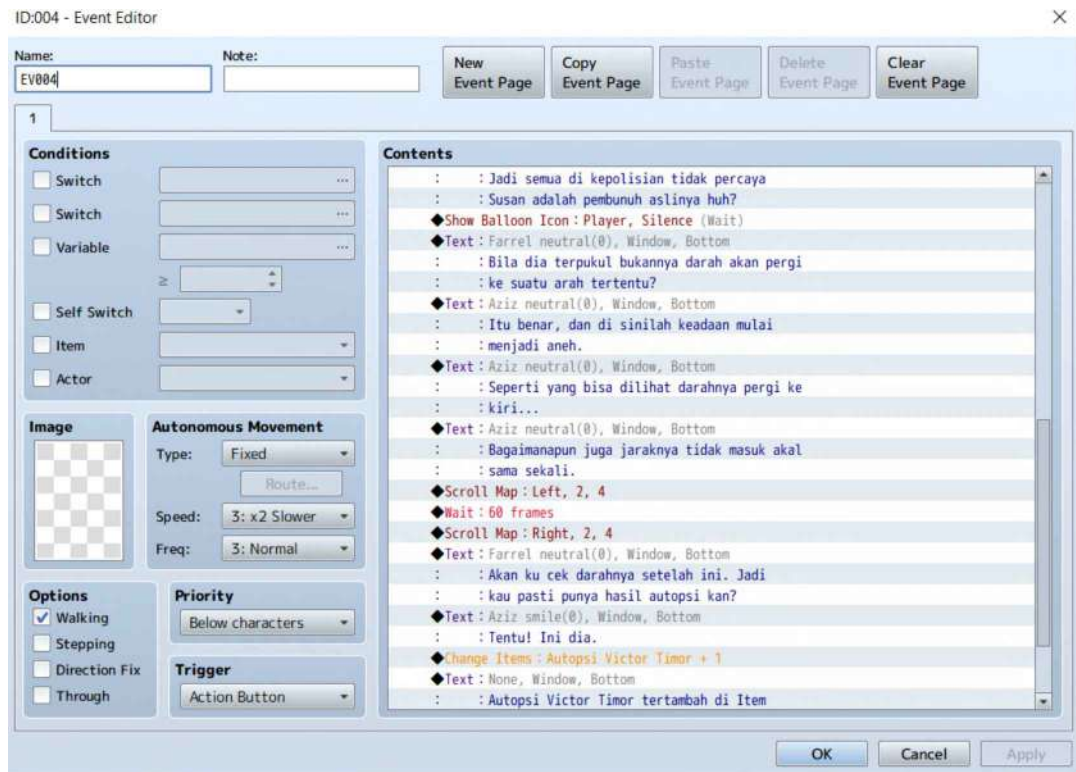
Gambar 5.72. Program TKP Bunda bagian 2



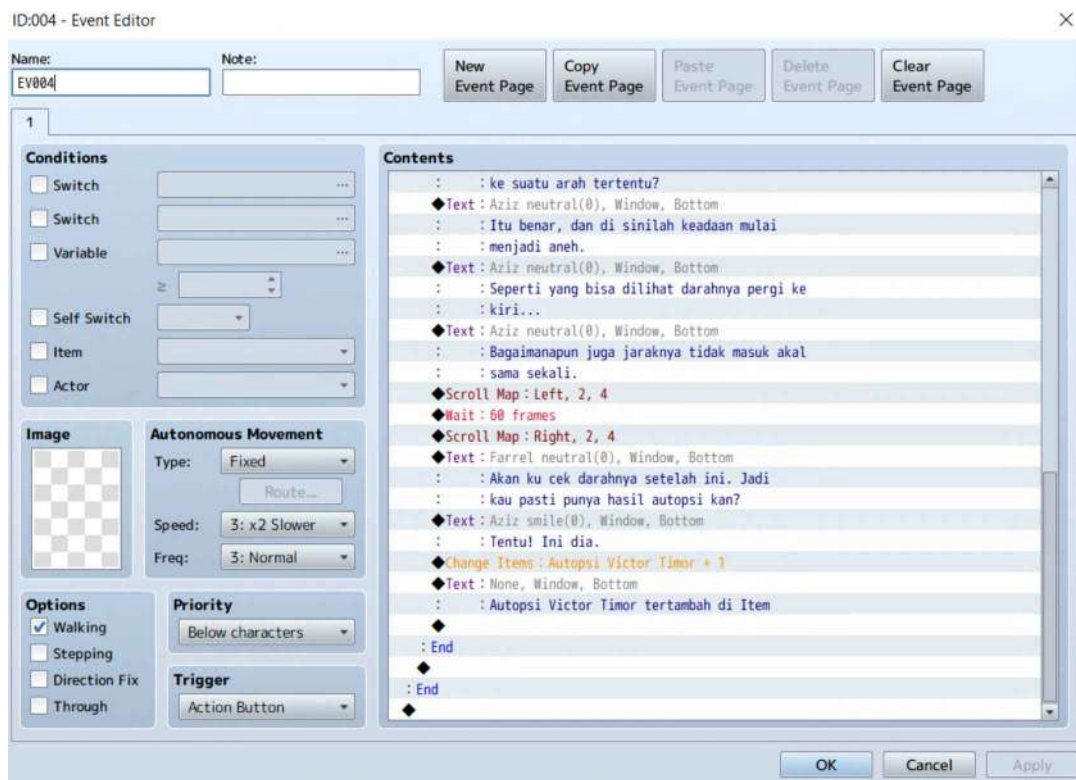
Gambar 5.73. Program TKP Bunda bagian 3



Gambar 5.74. Program TKP Victor bagian 1



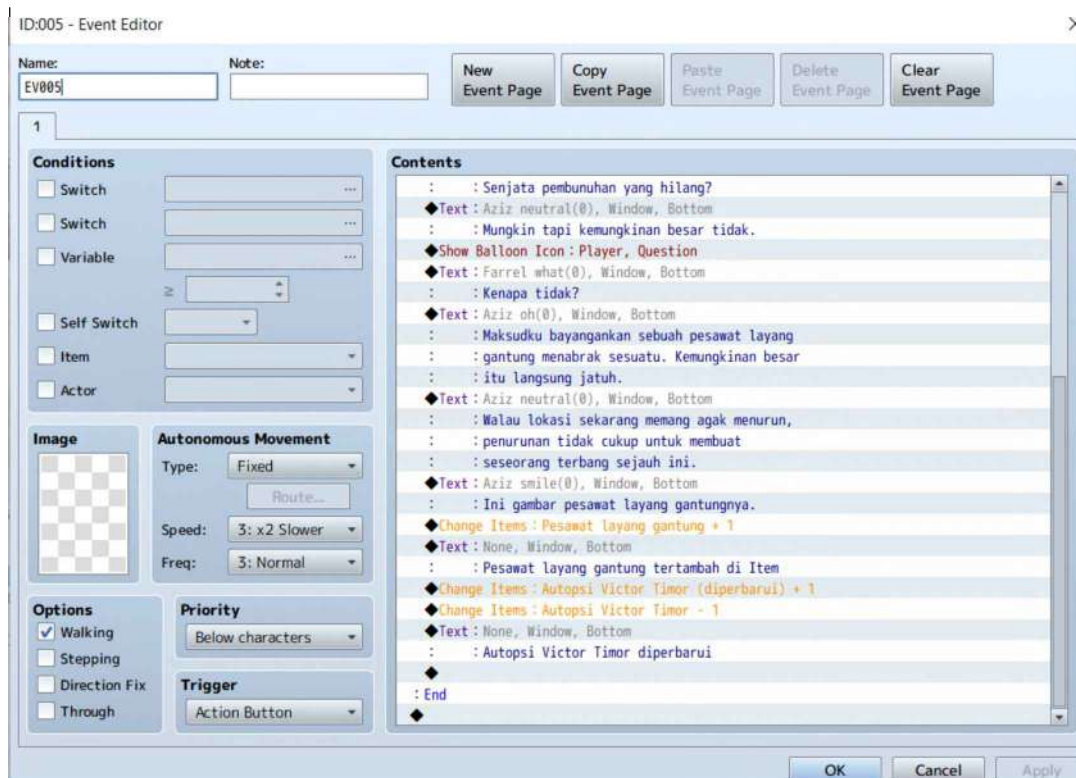
Gambar 5.75. Program TKP Victor bagian 2



Gambar 5.76. Program TKP Victor bagian 3

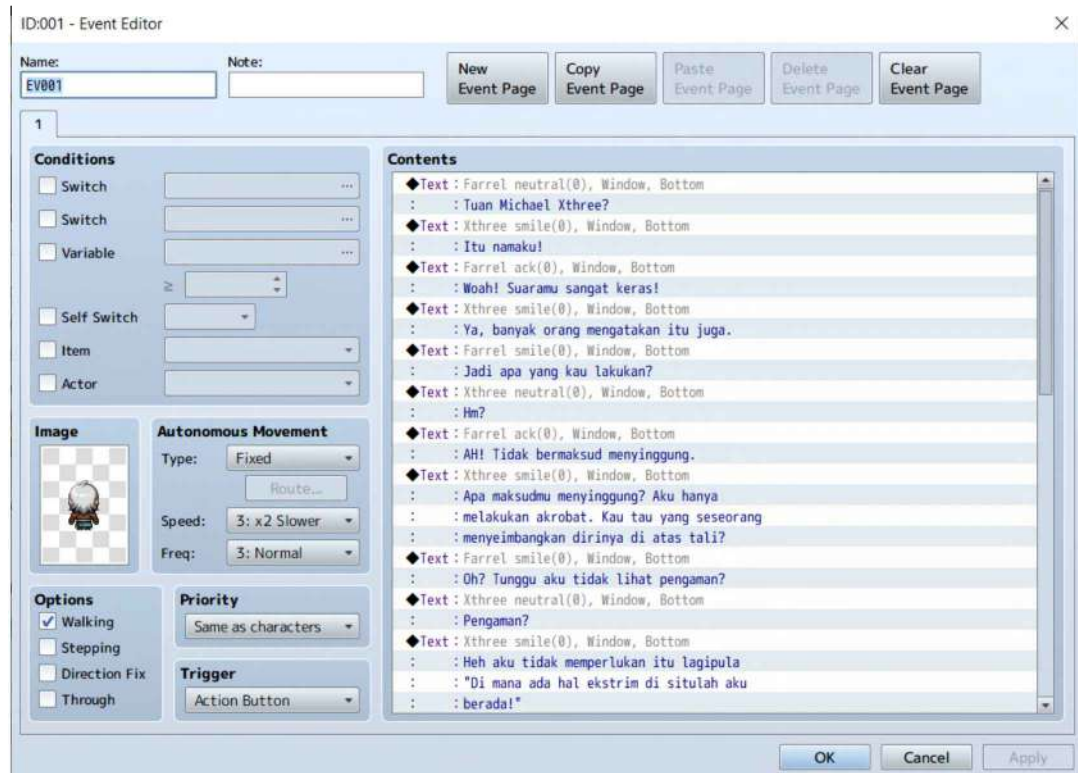


Gambar 5.77. Program TKP Victor (2) bagian 1

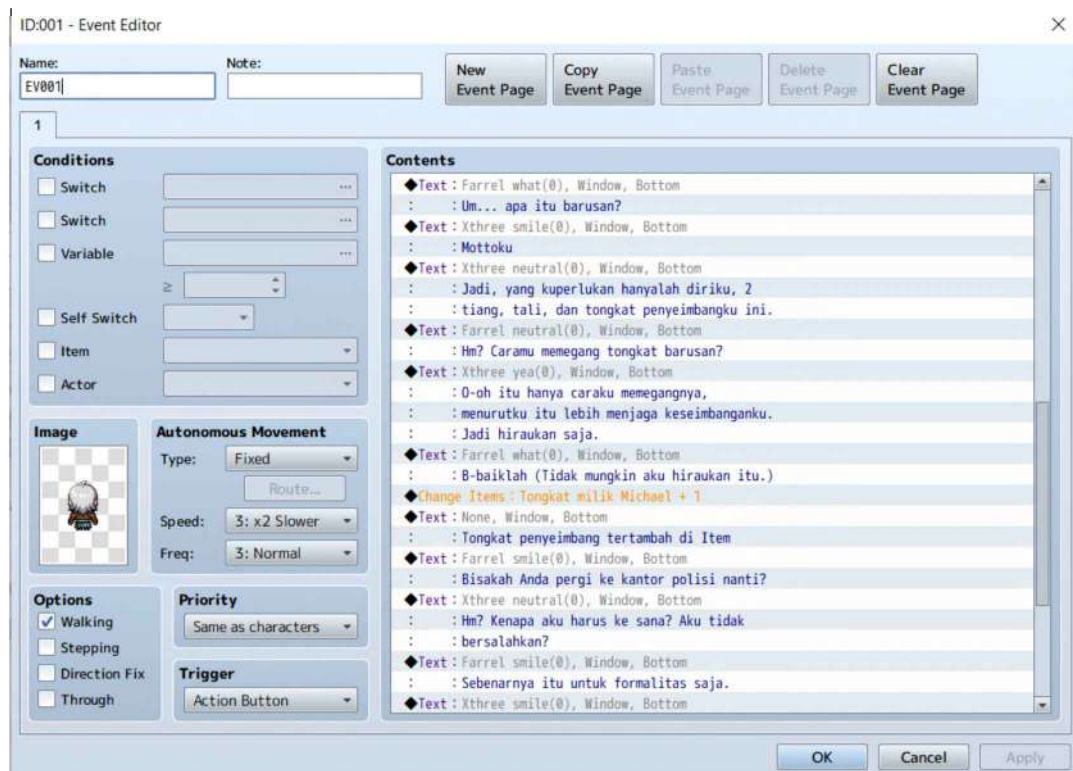


Gambar 5.78. Program TKP Victor (2) bagian 2

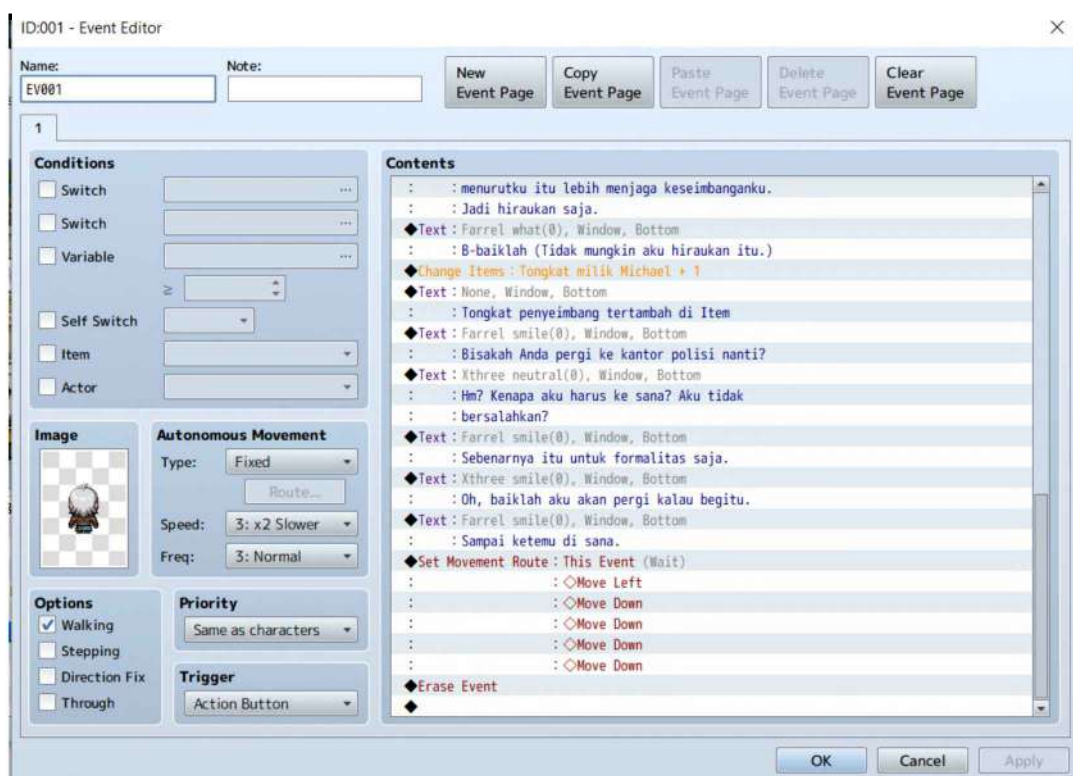
39. Selanjutnya *player* akan kembali ke *Bazaar* untuk bertemu dengan saksi. Maka berikut adalah perbedaan yang ada.



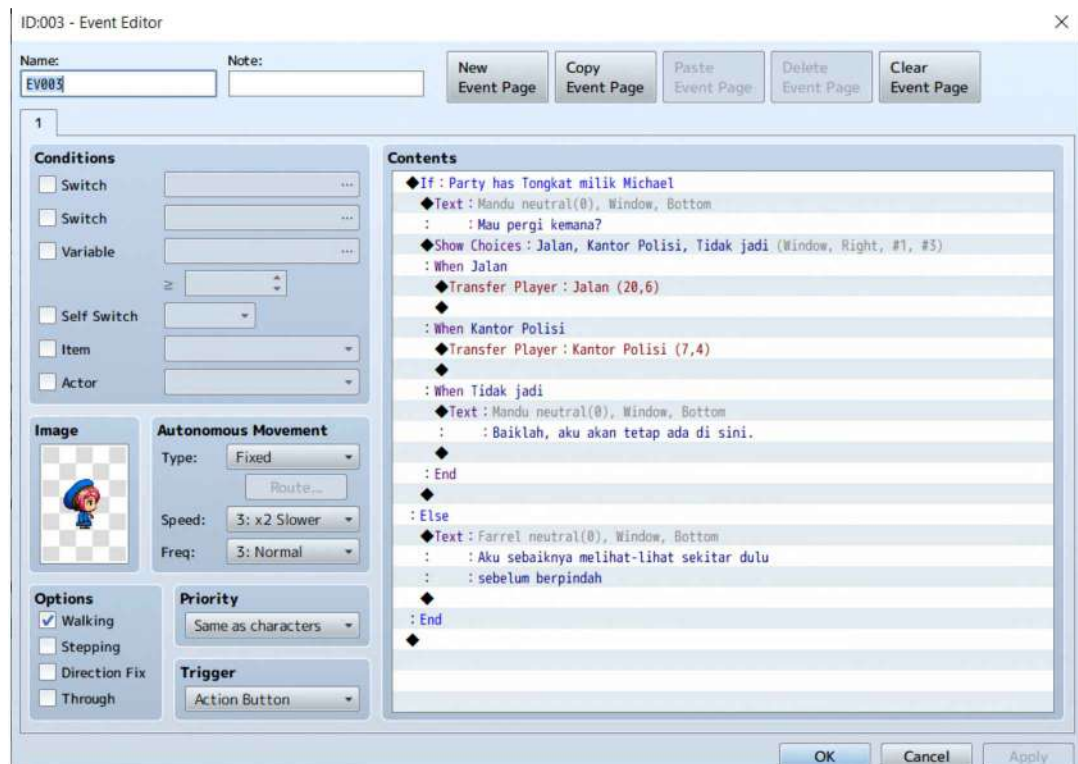
Gambar 5.79. *Program* interaksi dengan saksi bagian 1



Gambar 5.80. Program interaksi dengan saksi bagian 2

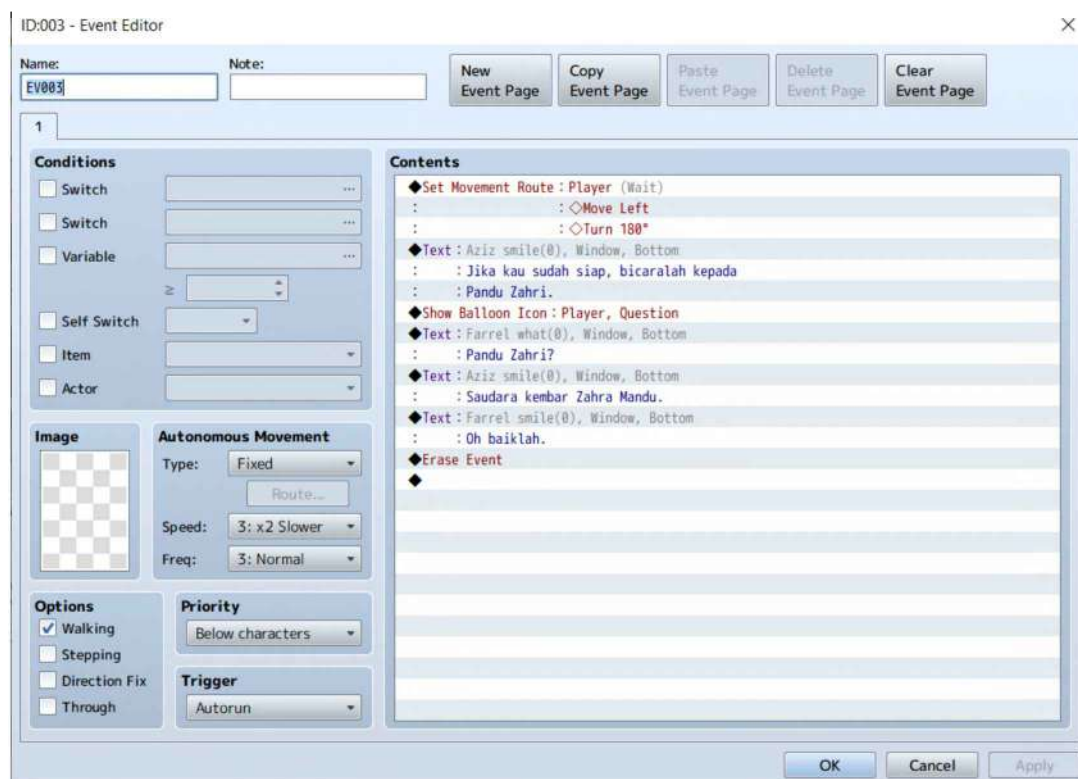


Gambar 5.81. Program interaksi dengan saksi bagian 3

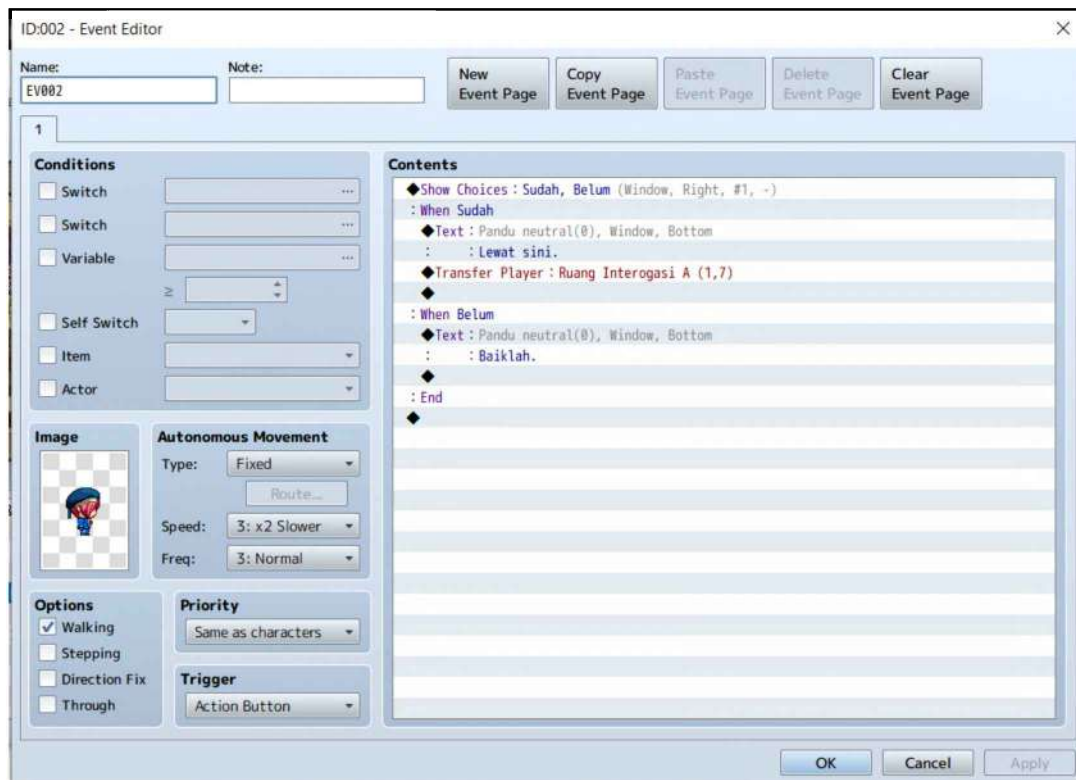


Gambar 5.82. Program perpindahan *player*

40. Berikut hasil *program* untuk *map* Kantor Polisi

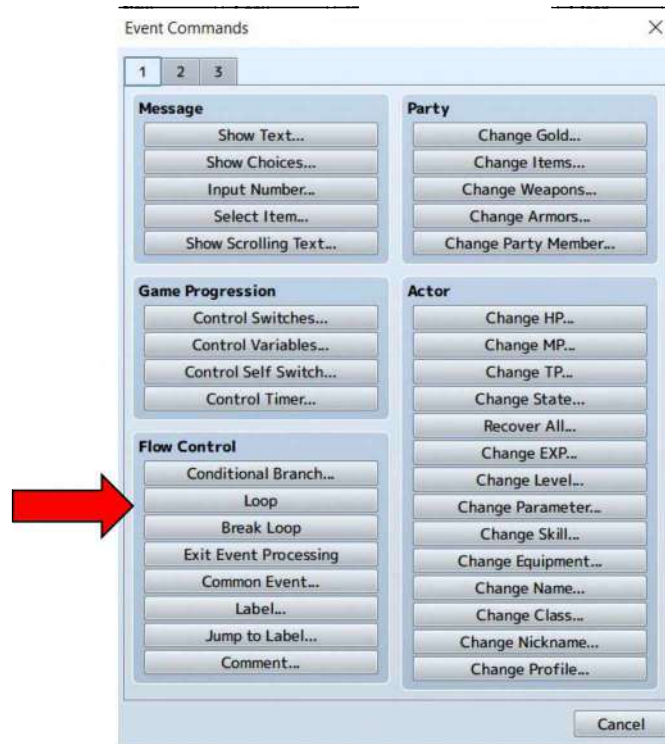


Gambar 5.83. *Program cutscene*



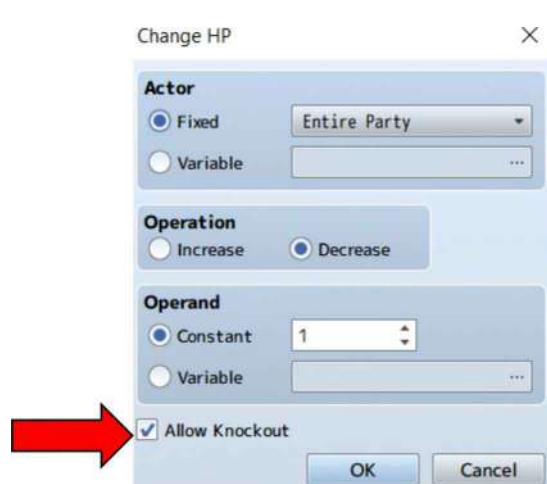
Gambar 5.84. *Program perpindahan player*

41. Untuk sesi interogasi akan digunakan *Loop* agar testimoni *character* terus berputar sampai *player* menemukan jawaban yang benar.



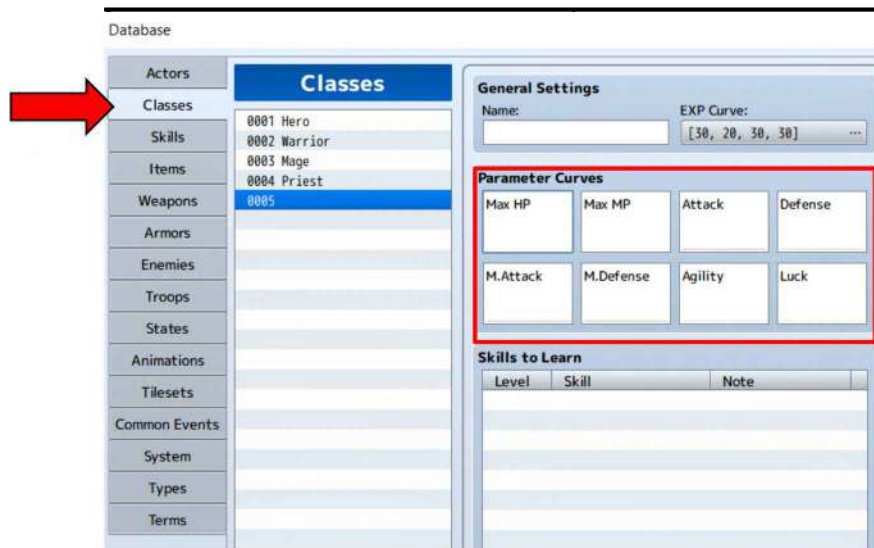
Gambar 5.85. Halaman 1 *Event Commands*

42. Selain penggunaan *Loop* di sesi interogasi “nyawa” *player* akan berkurang, untuk melakukan ini gunakan *Change HP*. Cara kerjanya sama seperti *Change Items* tetapi, di *Change HP* centang checkbox *Allow Knockout*. *Allow Knockout* memungkinkan untuk *player* mencapai nyawa 0 sedangkan bila tidak dicentang nyawa akan tersangkut pada 1.

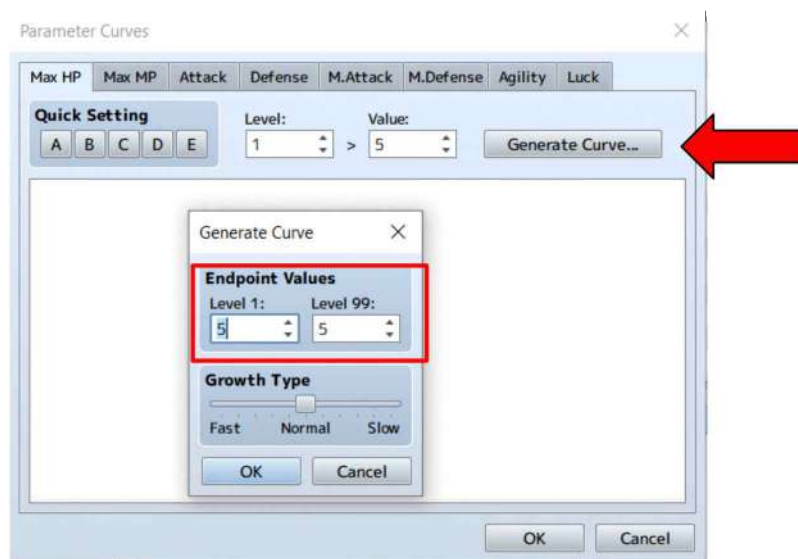


Gambar 5.86. Halaman *Change HP*

43. Karena “nyawa” *player* akan dibatasi 5 maka perlu mengganti nyawa *character* di *Database Classes*. klik dua kali pada *Max HP*, klik *Generate Curve* lalu ganti *Endpoint Values* menjadi 5.



Gambar 5.87. Halaman *Database Classes*



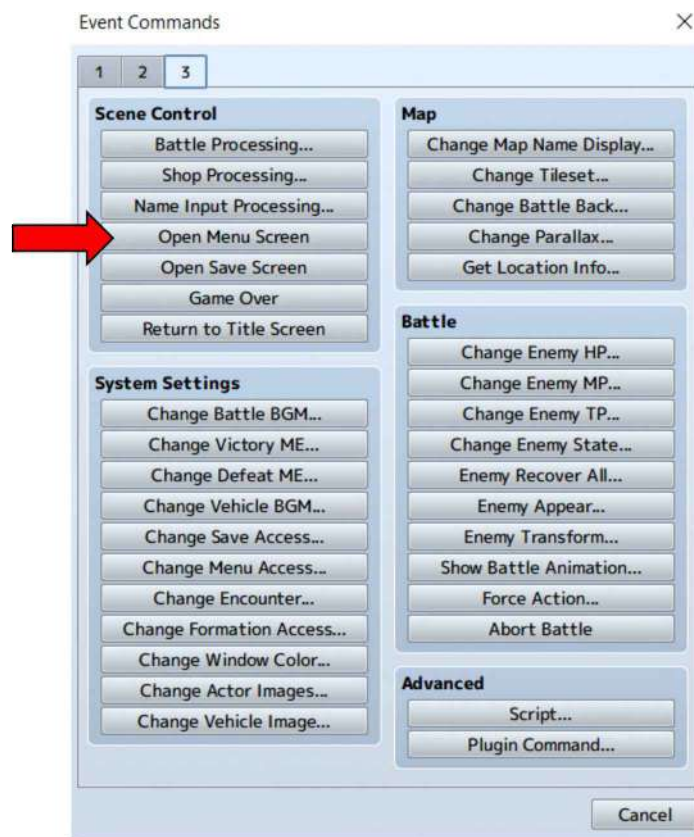
Gambar 5.88. Halaman *Generate Curve* pada *Parameter Curves*

44. Setelah itu gantilah *Class character* di *Database Actors*.



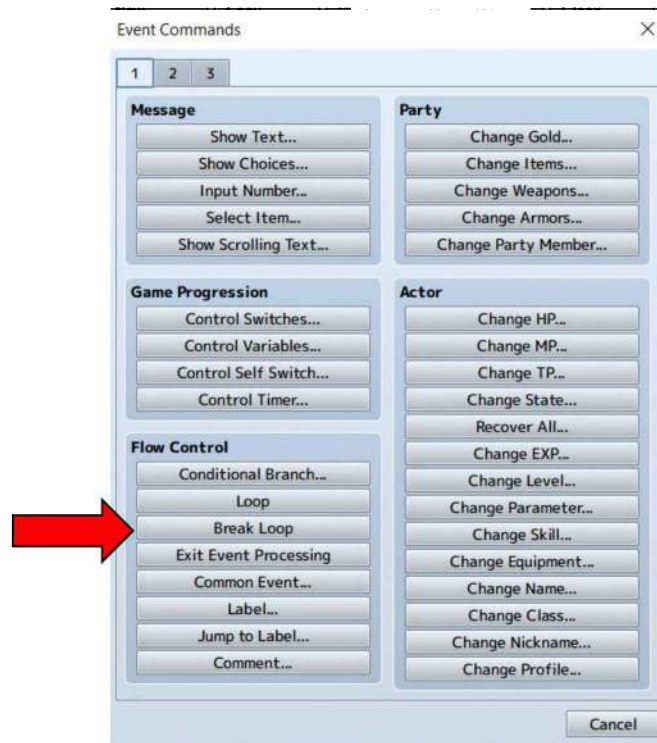
Gambar 5.89. Halaman *Database Actors*

45. Untuk membuka *menu* saat memilih pilihan, gunakan *Open Menu Screen*.



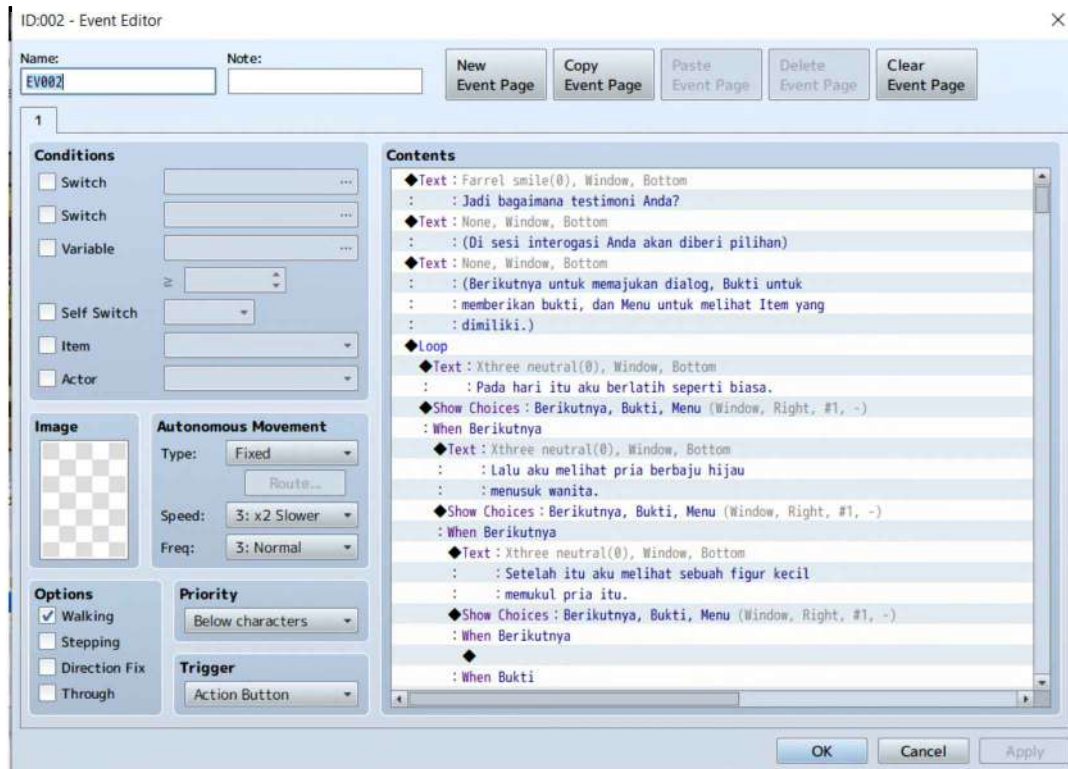
Gambar 5.90. Halaman 3 *Event Commands*

46. Bila *player* memberi bukti yang benar pada bagian testimoni yang benar maka *Loop* akan hilang dan lanjut ke testimoni yang lain. Untuk menghilangkan *Loop* gunakan *Break Loop*.

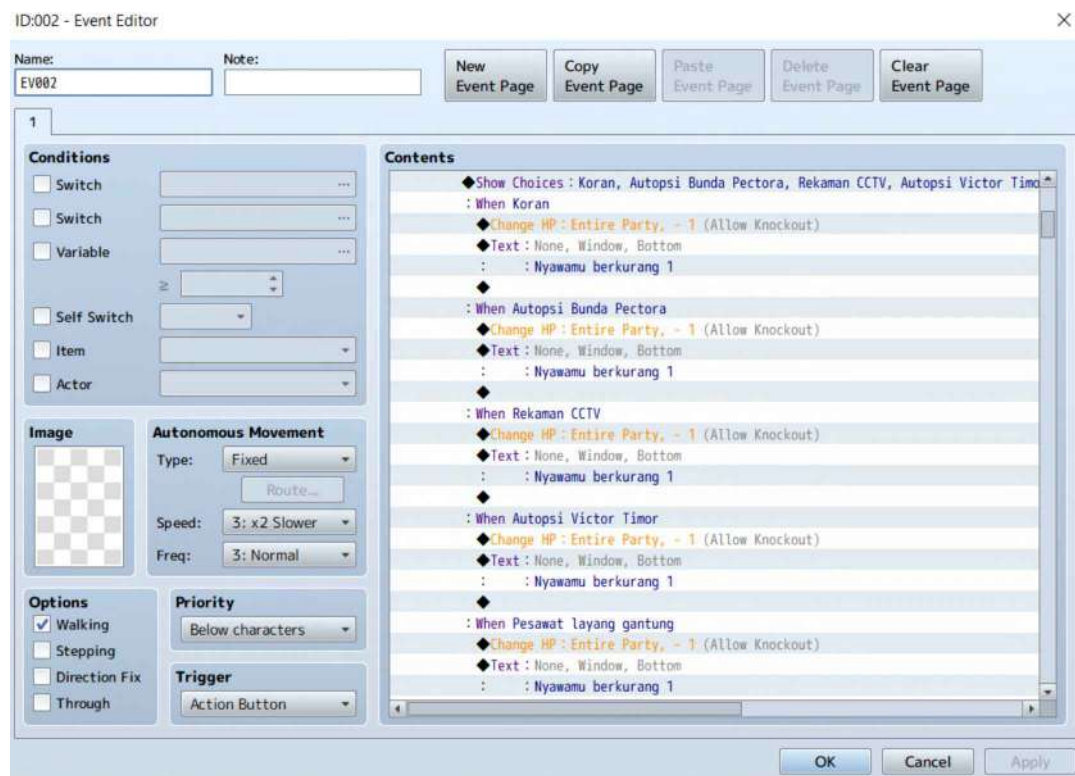


Gambar 5.91. Halaman 1 *Event Commands*

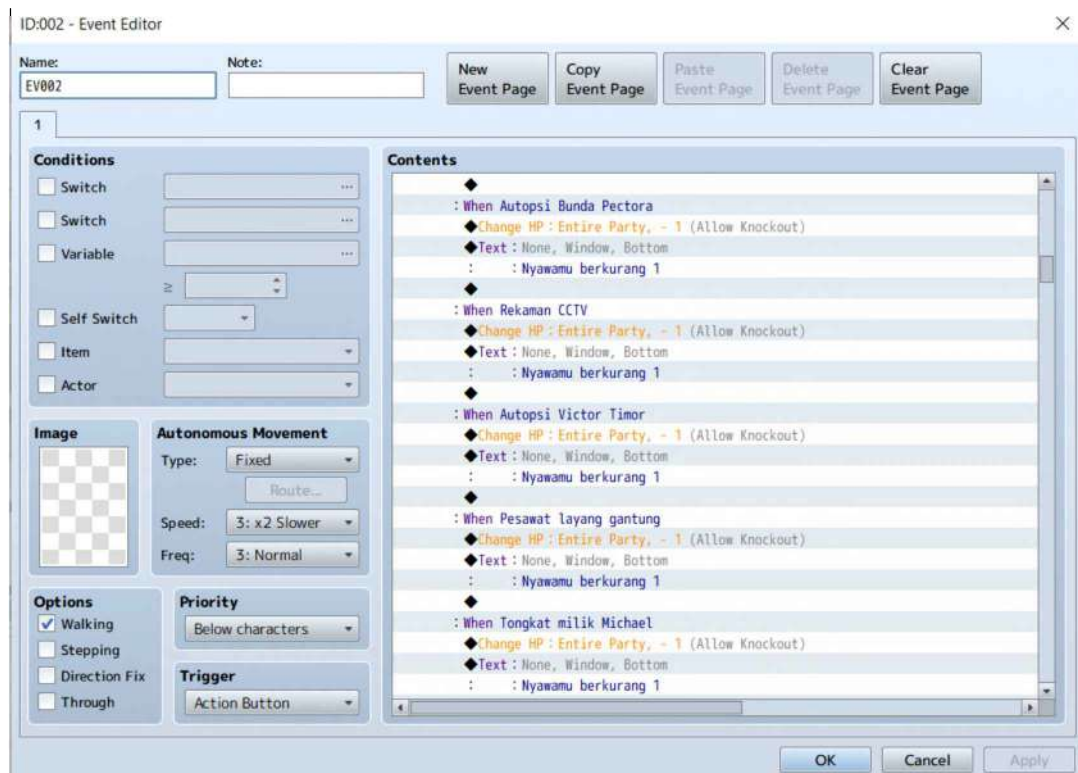
47. Berikut hasil *program* untuk *map* Ruang Interogasi



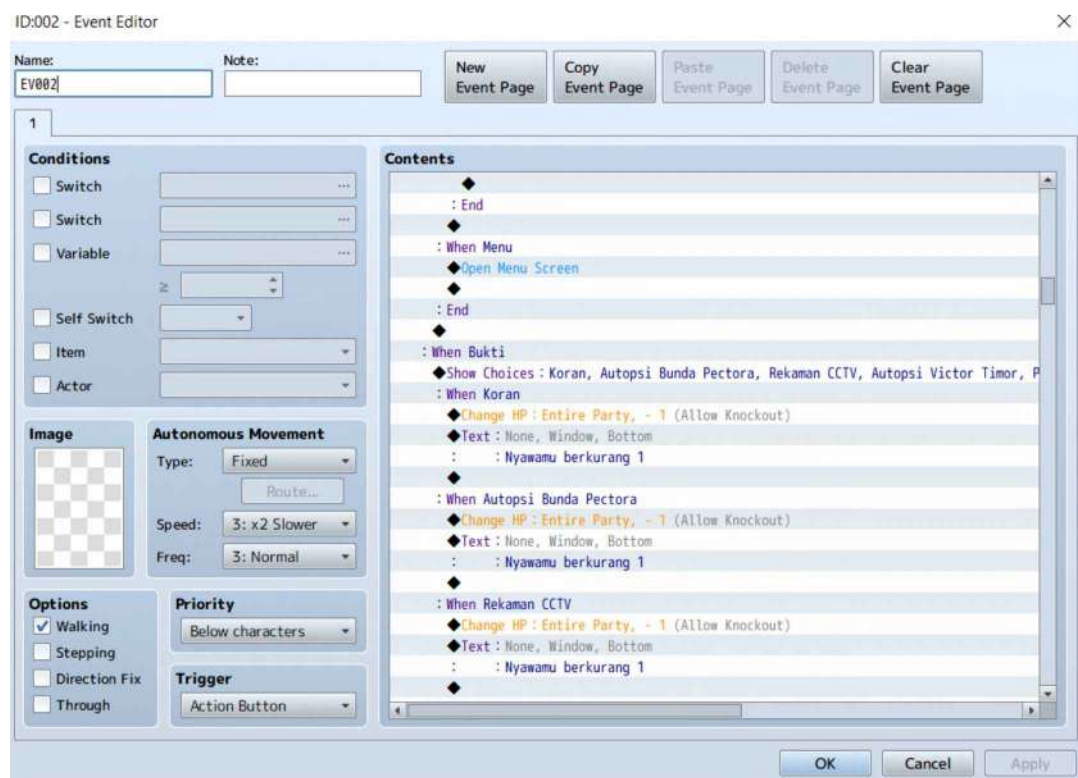
Gambar 5.92. Program interogasi bagian 1



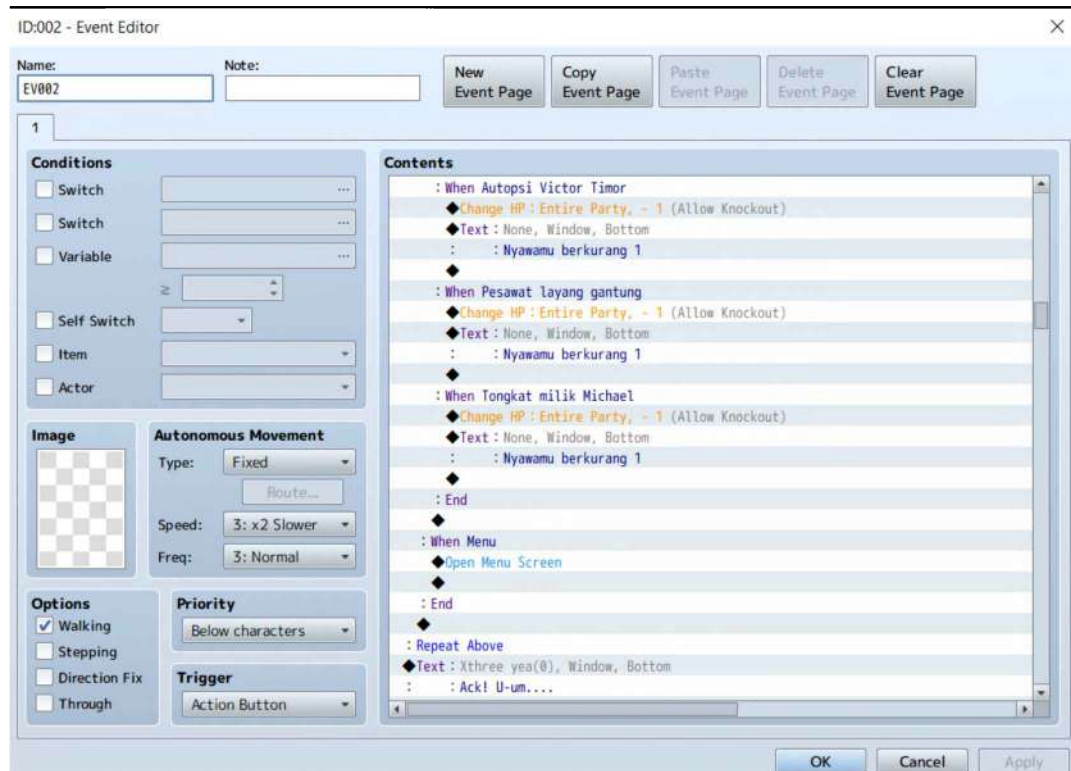
Gambar 5.93. Program interogasi bagian 2



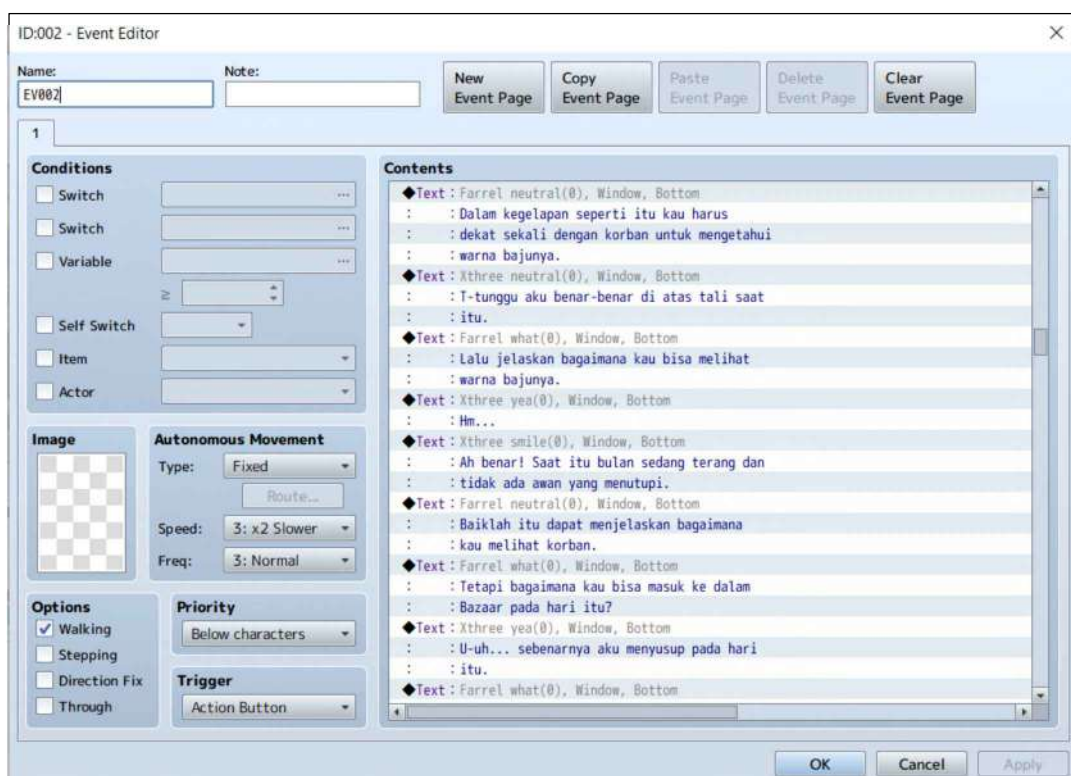
Gambar 5.94. Program interogasi bagian 3



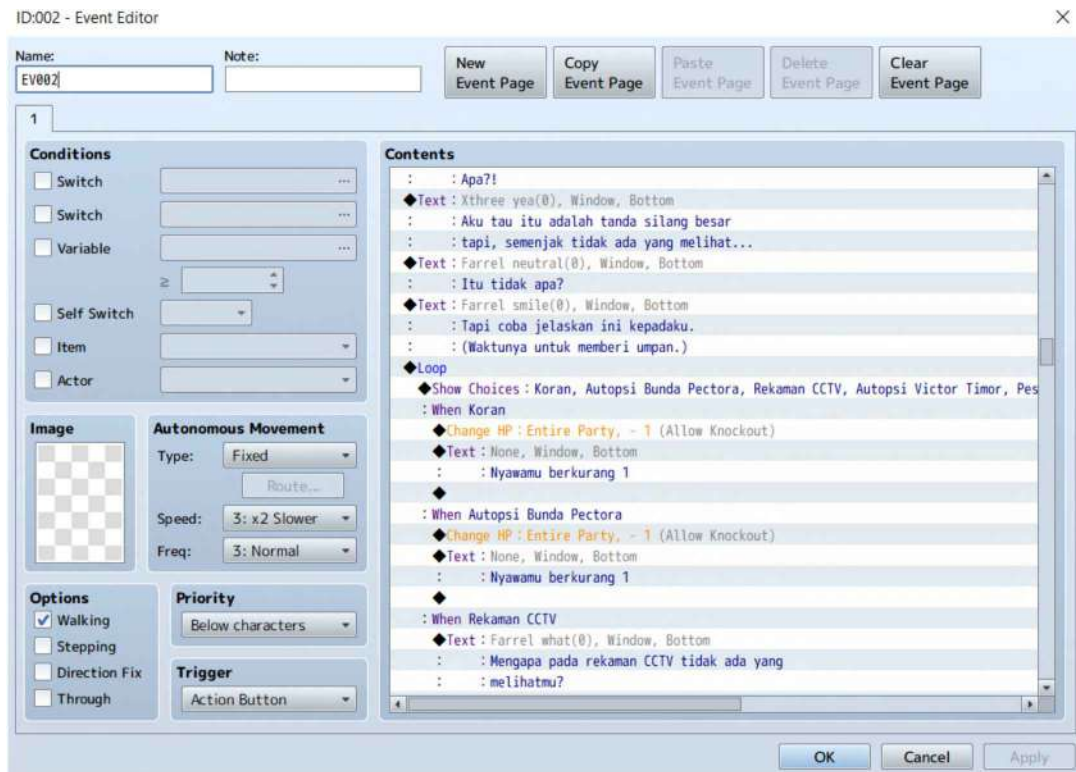
Gambar 5.95. Program interogasi bagian 4



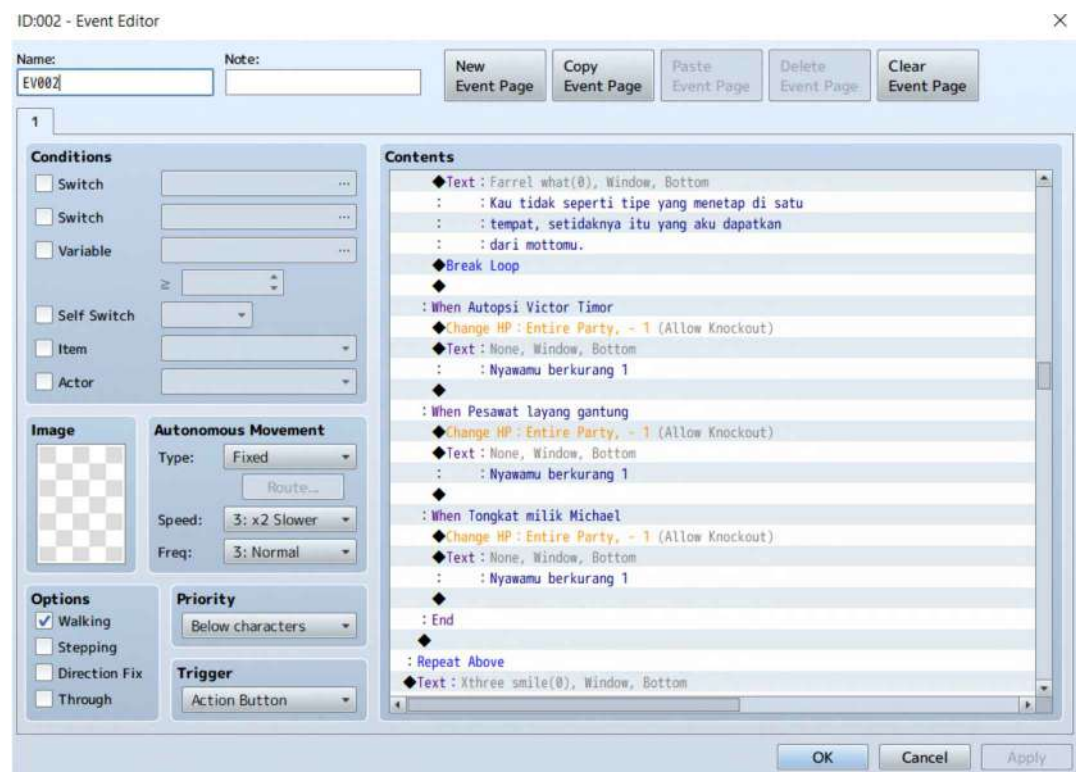
Gambar 5.96. Program interogasi bagian 5



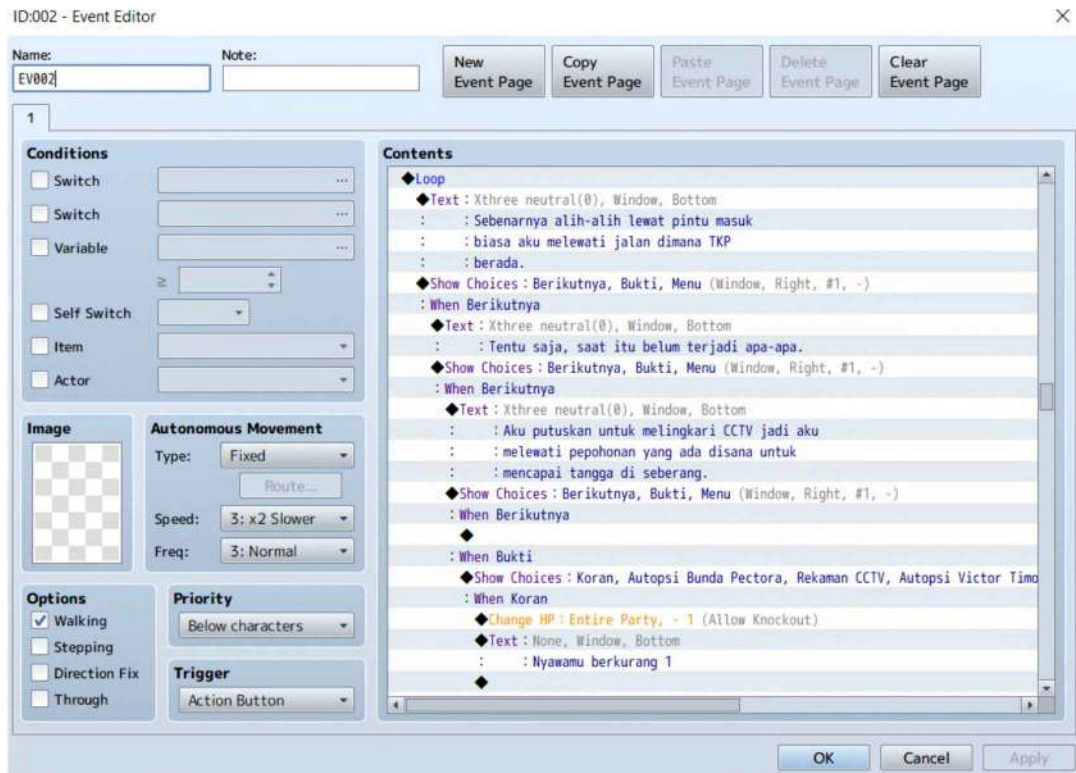
Gambar 5.97. Program interogasi bagian 6



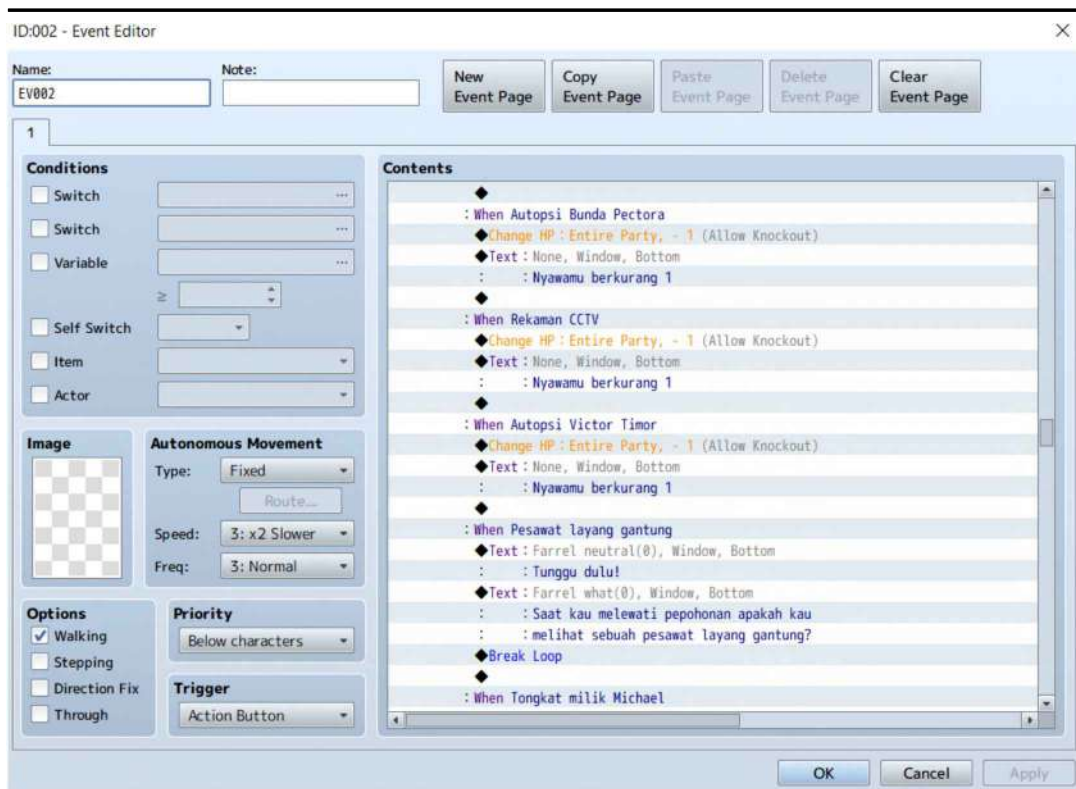
Gambar 5.98. Program interogasi bagian 7



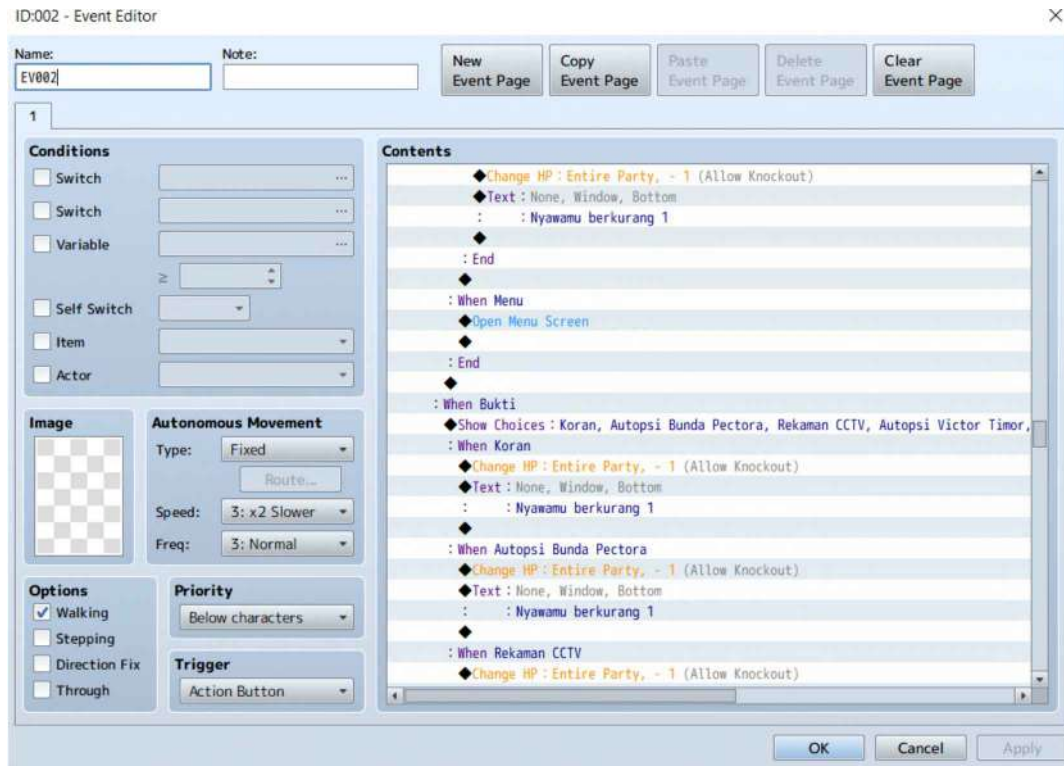
Gambar 5.99. Program interogasi bagian 8



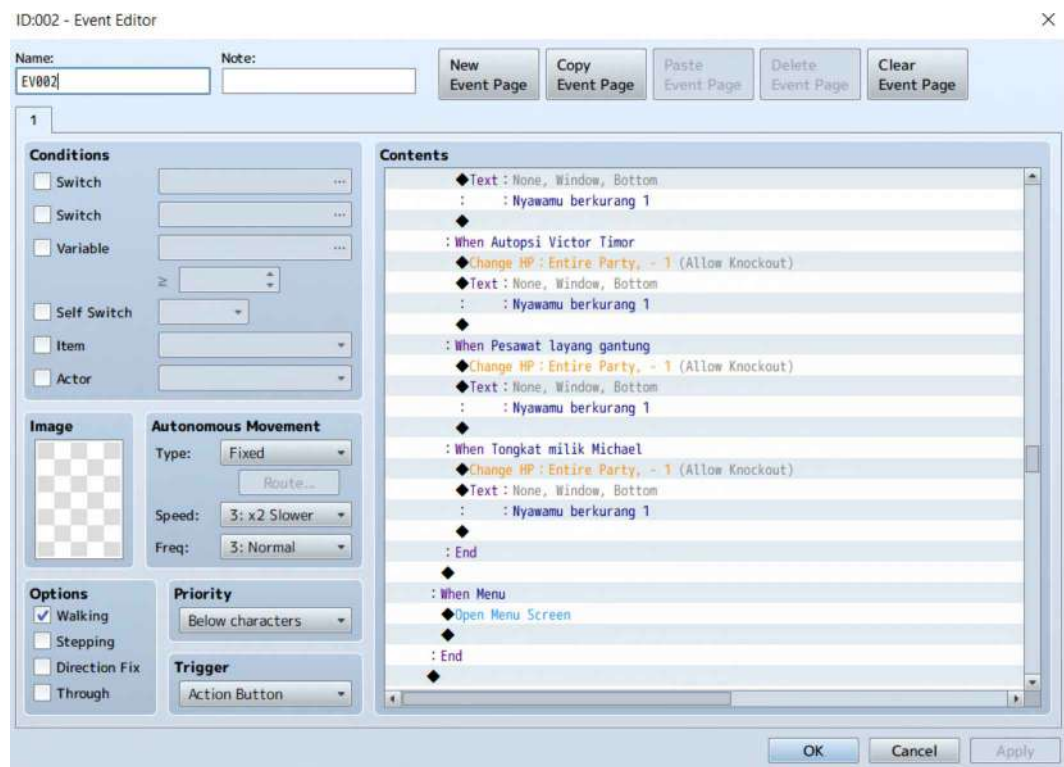
Gambar 5.100. Program interogasi bagian 9



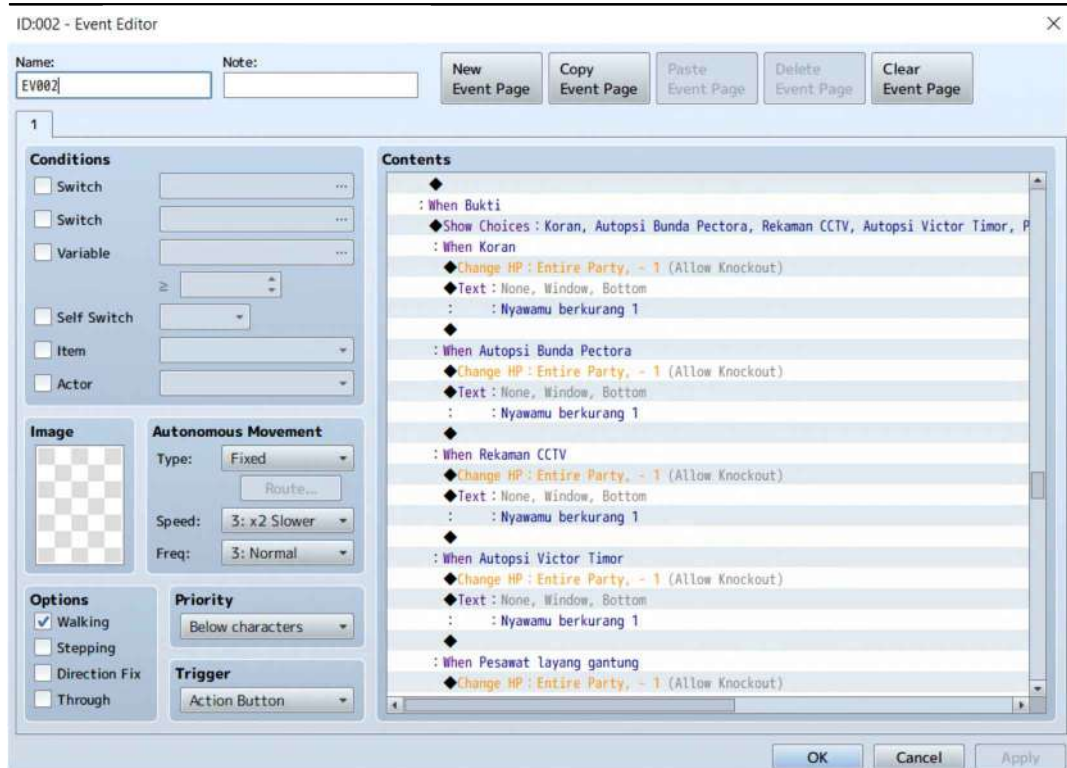
Gambar 5.101. Program interogasi bagian 10



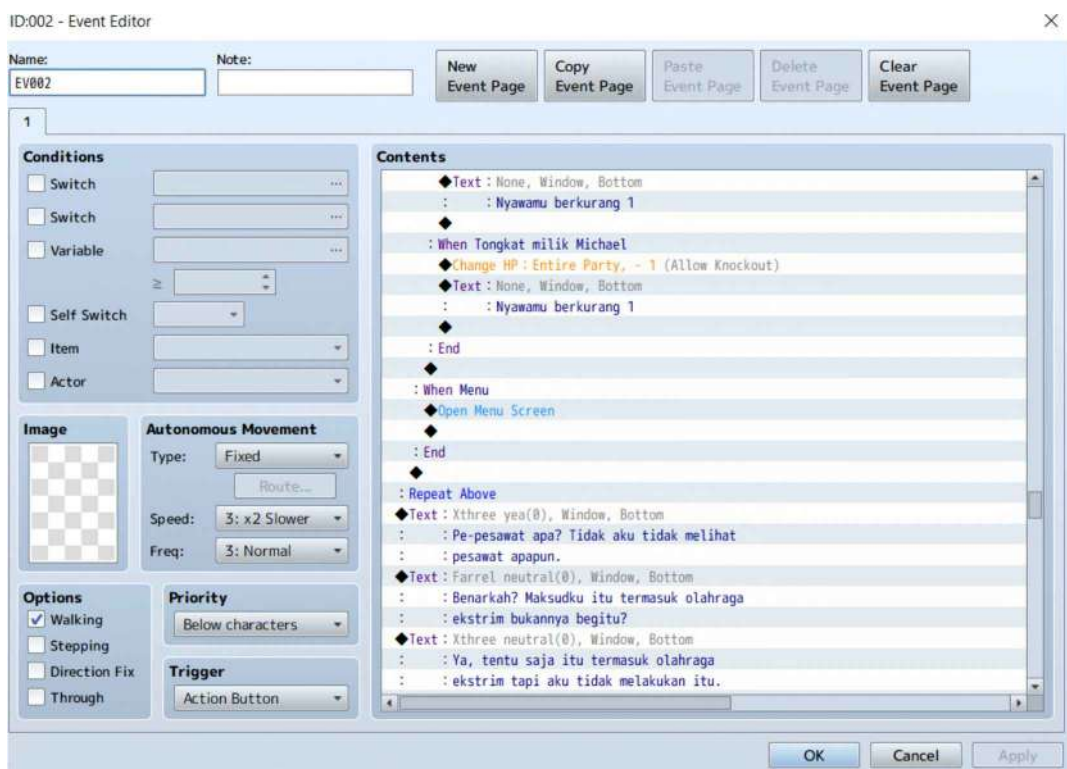
Gambar 5.102. Program interogasi bagian 11



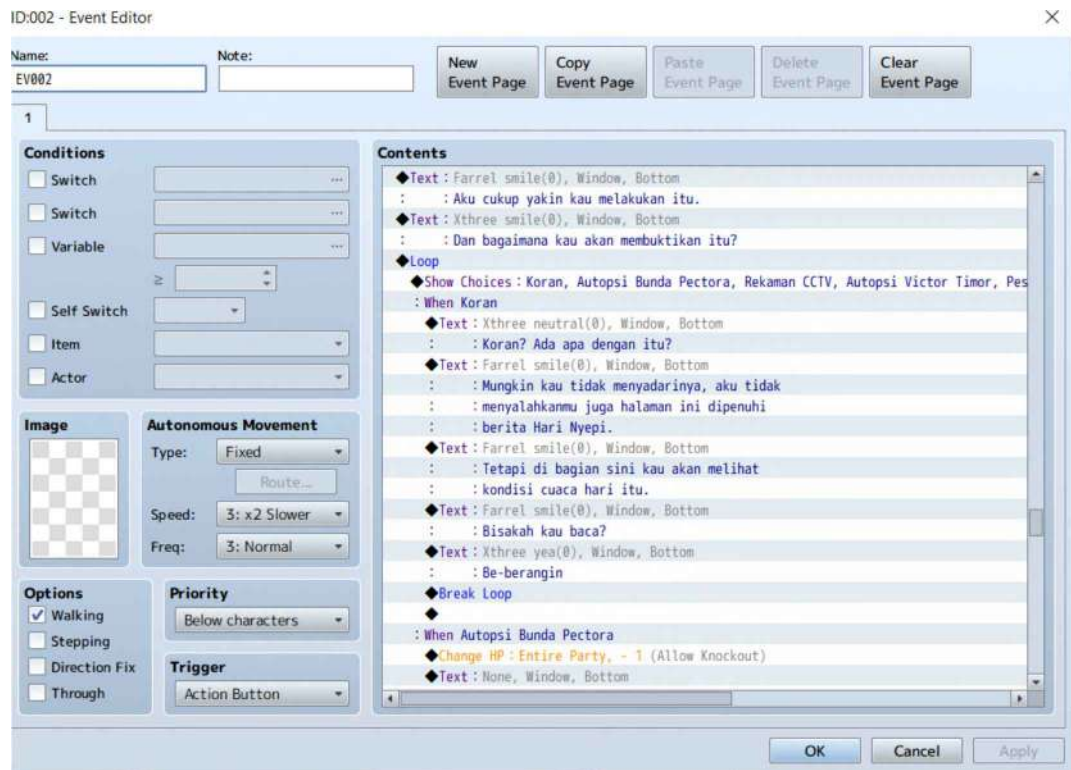
Gambar 5.103. Program interogasi bagian 12



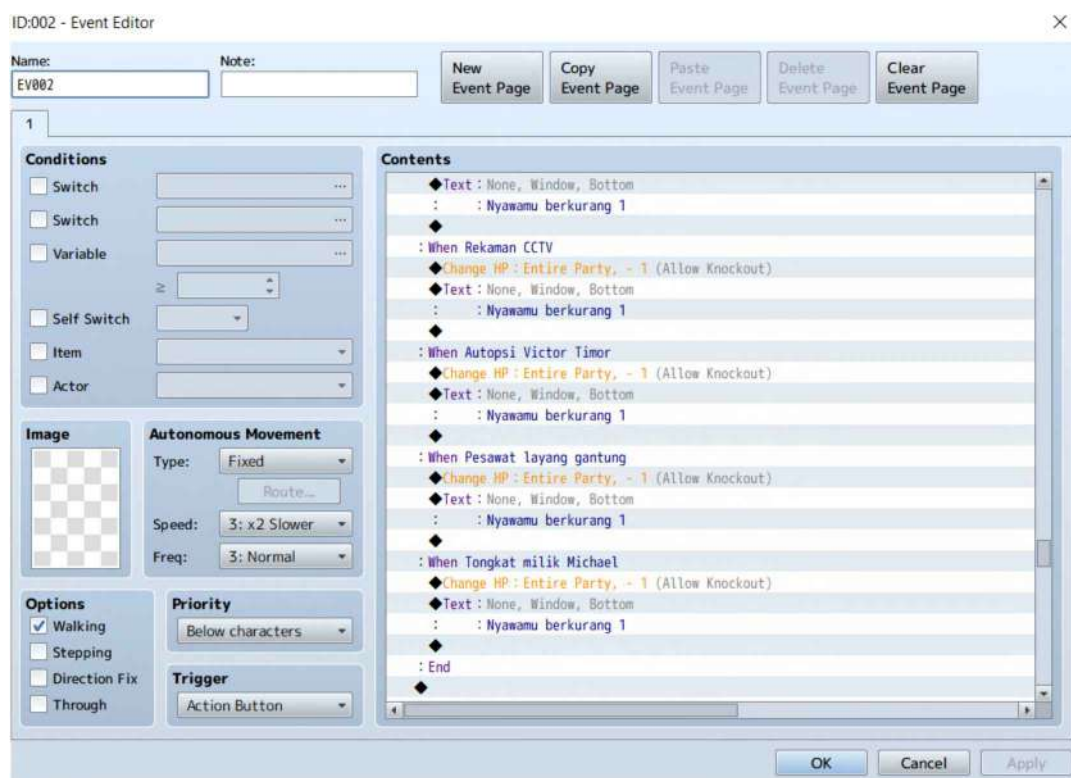
Gambar 5.104. Program interogasi bagian 13



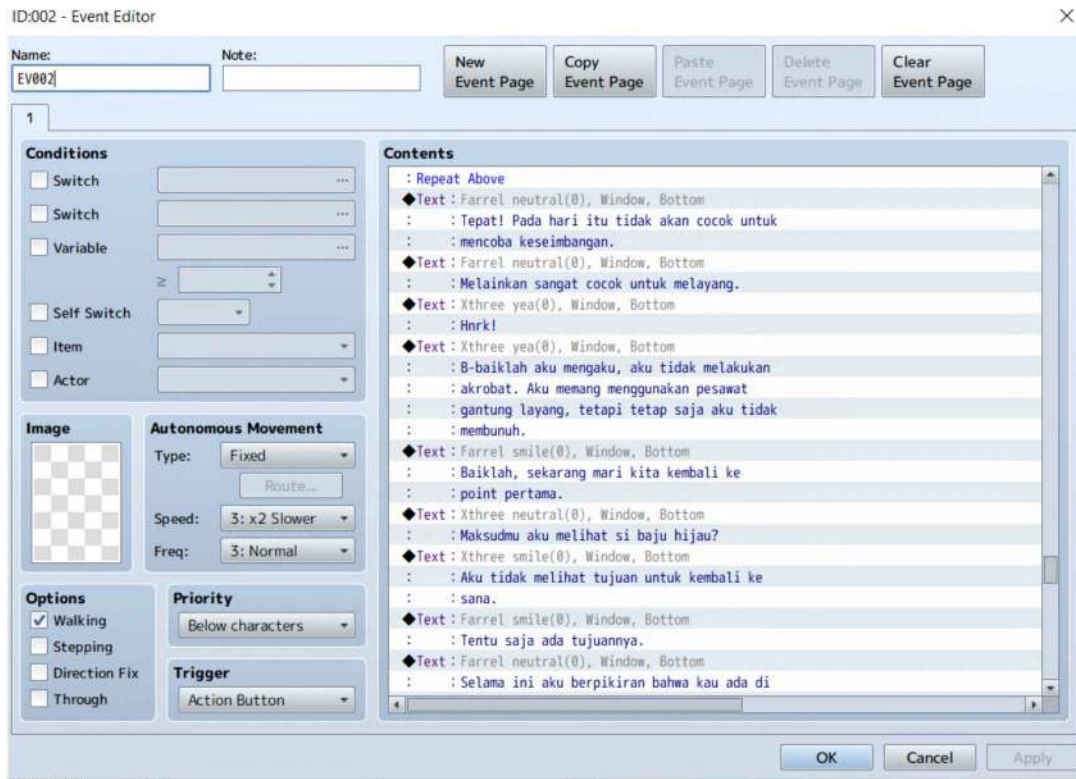
Gambar 5.105. Program interogasi bagian 14



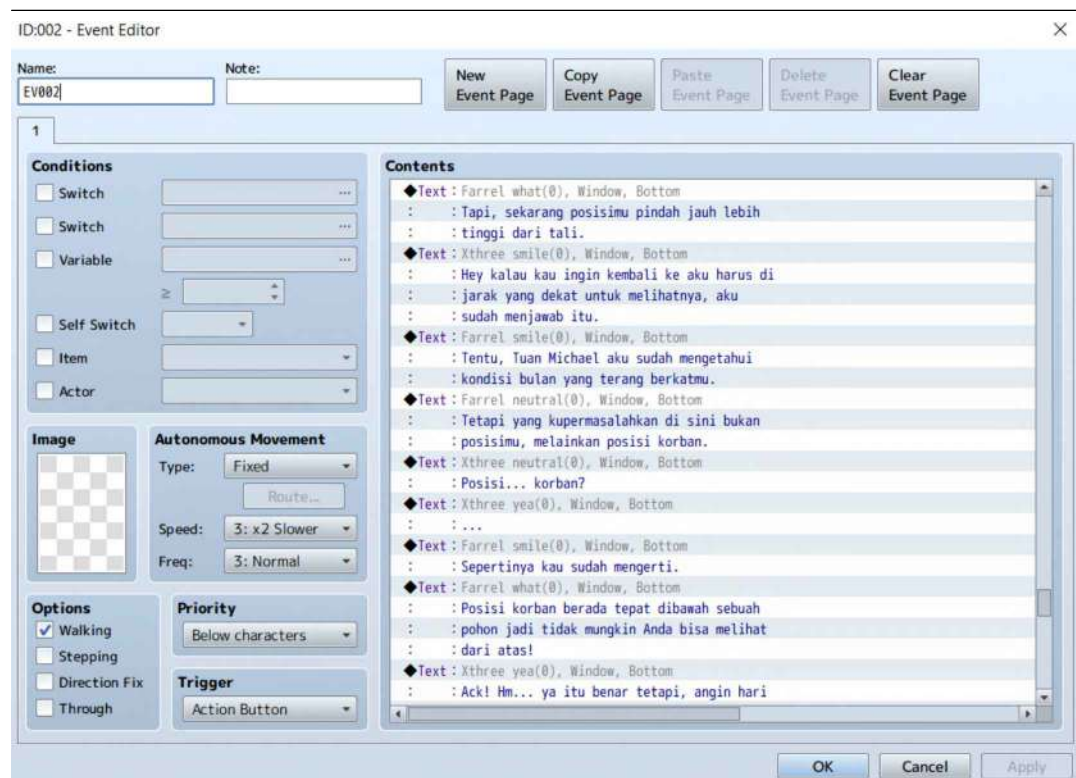
Gambar 5.106. *Program* interogasi bagian 15



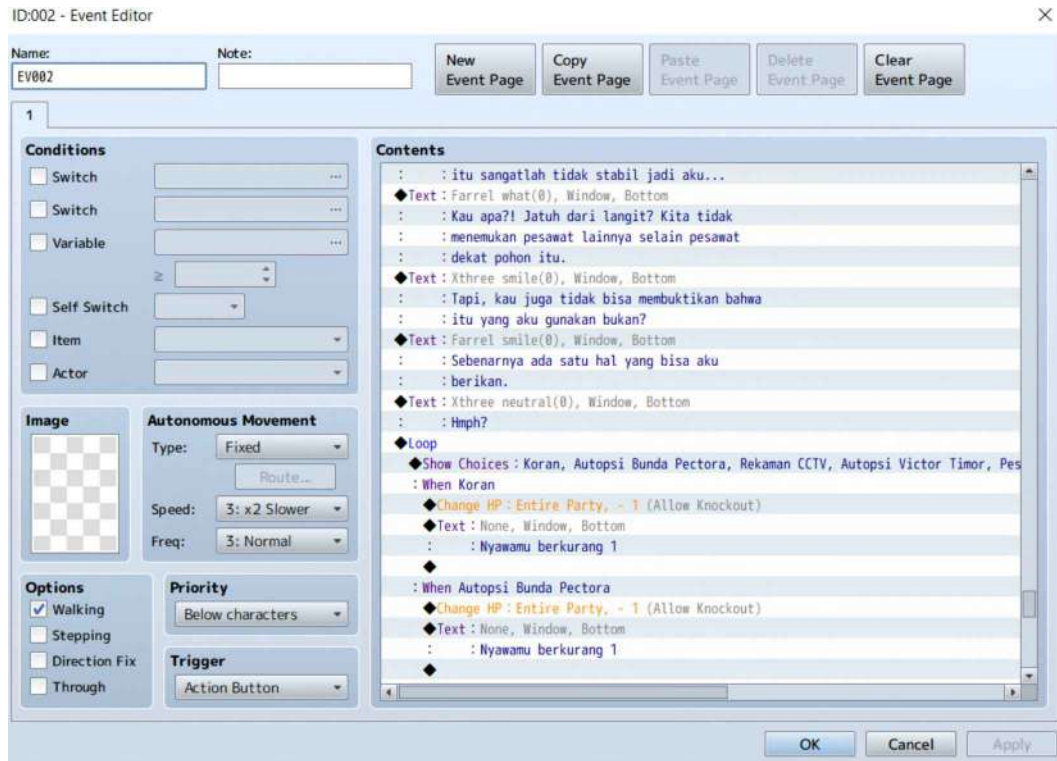
Gambar 5.107. *Program* interogasi bagian 16



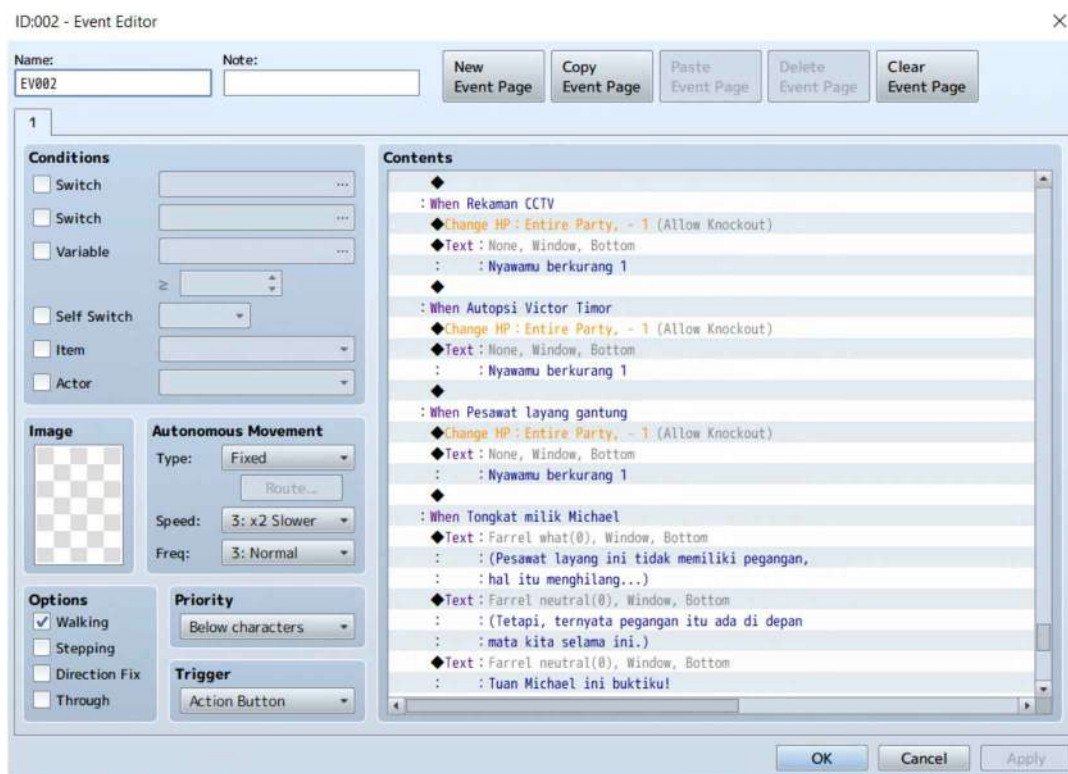
Gambar 5.108. Program interogasi bagian 18



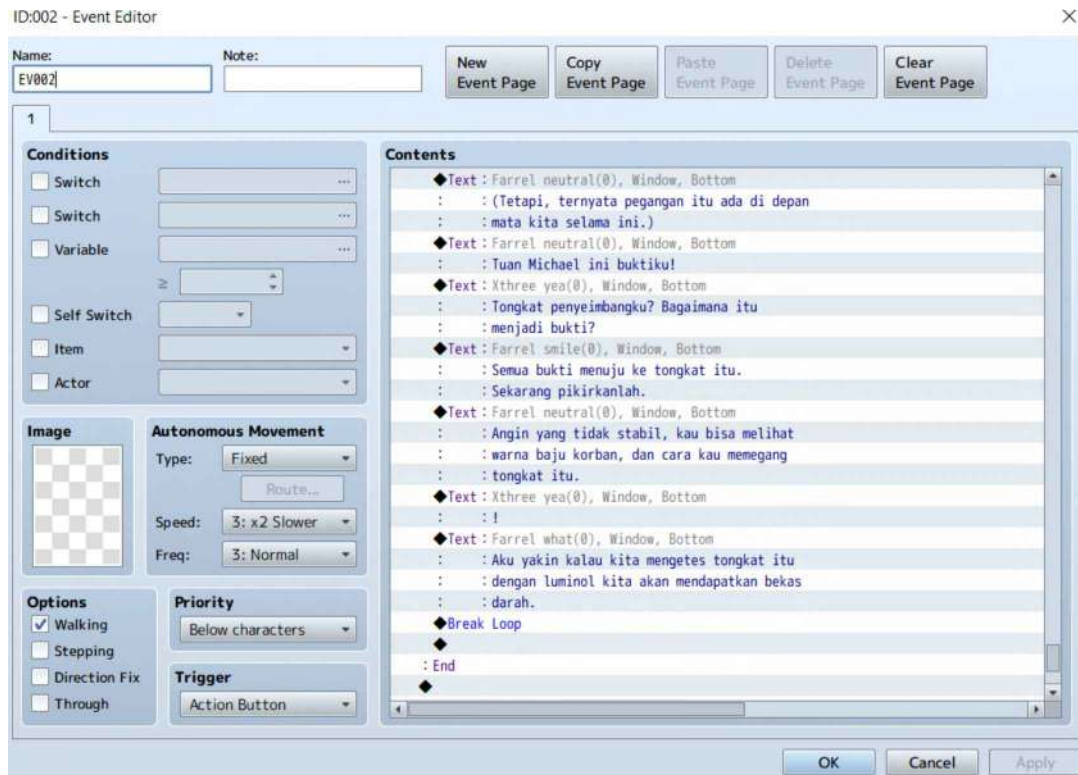
Gambar 5.109. Program interogasi bagian 19



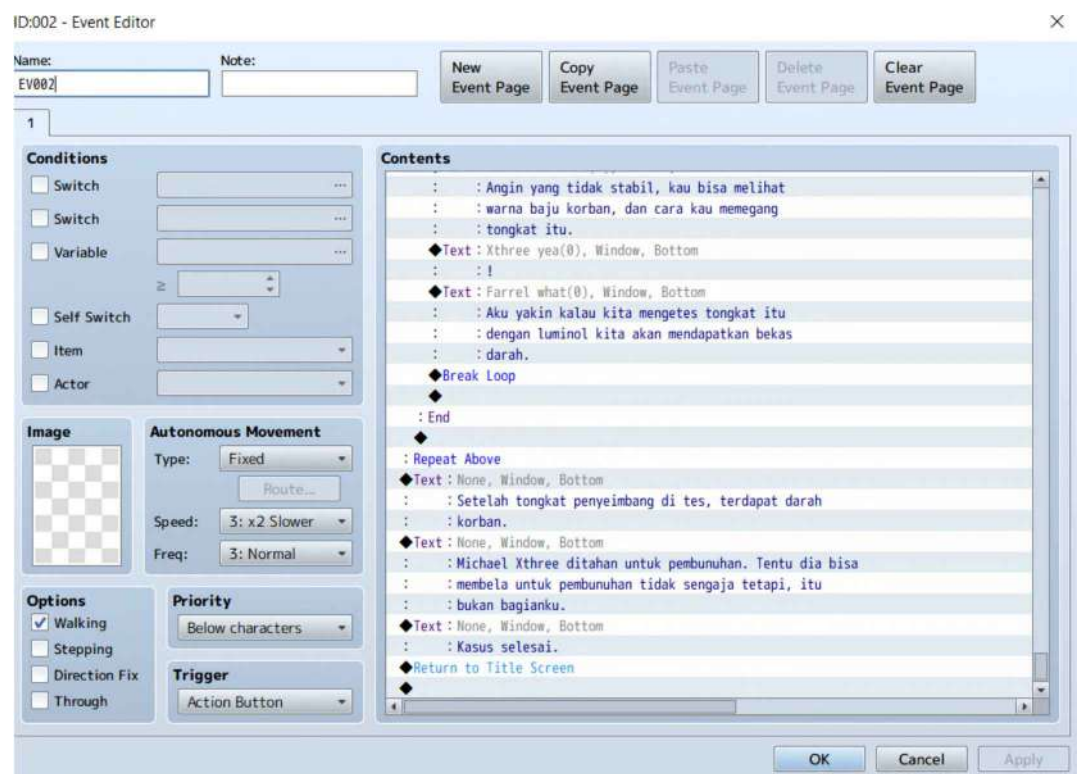
Gambar 5.110. Program interogasi bagian 20



Gambar 5.111. Program interogasi bagian 21



Gambar 5.112. Program interogasi bagian 22



Gambar 5.113. Program interogasi bagian 23

Link game

<https://drive.google.com/file/d/1L6k8gcTr-EF8YaC0j2kBqJtCK5EkFnbT/view?usp=sharing>

Petualangan Pelajar

Marcello Jovan Christopher Kalbuadi

Pada zaman dahulu, ada suatu desa yang belum diketahui namanya. Para warga di desa ini awalnya semua hidup dengan damai tanpa adanya kepemimpinan dari seorang raja. Tetapi suatu hari, datanglah King Julien dengan angkuhnya menghancurkan desa yang Harold dan Julia tinggali. Awalnya Harold hanya mematuhi apa yang dikatakan King Julien karena dia melihat konsekuensi yang diberikan jika tidak patuh dengannya. Harold pun melihat Julia dibawa karena tidak patuh ke King Julien, melihat hal ini pun Harold sedih karena temannya dibawa. Lalu tekad Harold untuk menyelamatkan temannya pun menjadi tinggi dan dia berniat untuk membebaskan temannya dari tawanan King Julien

TANTANGAN

Tantangan pada game ini adalah diharuskan melawan musuh sembari mencari informasi dengan cara bertanya pada cara bertanya kepada para warga di desa yang belum diketahui namanya. Untuk dapat menyelamatkan temannya pun tidak mudah karena harus melewati tempat-tempat yang tidak tahu di depan pemain ada apa atau tantangan apa yang harus dilewati sehingga harus berhati-hati dalam menyusuri tiap petanya.

KARAKTER GAME

Karakter pada *game* yang akan saya jelaskan disini merupakan karakter utamanya saja.



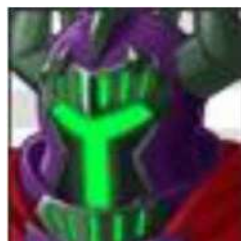
Gambar 6.1. Harold

Namanya Harold, dia adalah seorang pelajar di kota yang belum diketahui namanya untuk menyelamatkan temannya yang ditawan oleh King Julien yaitu penguasa Kerajaan Kegelapan.



Gambar 6.2. Julia

Julia adalah teman Harold yang ditawan oleh King Julien karena memberontak saat King Julien ingin menguasai desanya.



Gambar 6.3. King Julien

King Julien adalah penguasa kerajaan kegelapan yang mempunyai ambisi untuk menguasai desa yang Harold tinggal dan dia juga dikenal sangat kejam untuk orang yang tidak sepaham dengan dia.

Peta



Gambar 6.4. Desa pertama kali *spawn*

Pada *level* ini, pemain diharuskan menjawab pertanyaan dari para warga untuk bisa mendapatkan informasi penting mengenai jalan untuk menuju kerajaan kegelapan. Pertanyaan disini diambil dari banyak pelajaran tetapi tidak terlalu susah.



Gambar 6.5. Hutan tempat bertarung

Pada *level* ini, pemain juga diharuskan bertarung agar mendapatkan informasi dari para musuh yang kalah dan dapat berjalan dengan tenang tanpa adanya gangguan dari musuh.



Gambar 6.6. Depan kerajaan kegelapan

Di *level* ini, hanya akan berjalan melalui depan dari kerajaan kegelapan ini. Tidak ada musuh atau tantangan di *level* ini.



Gambar 6.7. Dalam kerajaan kegelapan

Pada *level* ini, pemain diharuskan melawan King Julien untuk menyelamatkan teman kita. Jika berhasil maka pemain akan menang. Jika pemain kalah dapat mengulangi lagi.

TAHAPAN PEMBUATAN GAME

Untuk tahap pertama dalam pembuatan *game* ini yang terpenting adalah karakter dan peta terlebih dahulu. Cara membuat dua unsur ini pun sangat *simple*. Jika kita ingin membuat karakter sendiri maka kita bisa menggunakan *character generator*. Klik ikon seperti pada gambar di bawah

ini.



Gambar 6.8. Tampilan utama RPG Maker MV

Pada opsi ini, pemain dapat membuat atau meng-
kustomisasi karakter, ikon dengan keinginan para
pemain. Ikon Gear

Jika Resource sendiri pun dapat memakai
karakter, Peta
Menu
yang sudah Peta kan dari game dengan klik ikon *gear*
seperti pada gambar 6.8.

Setelah klik pada ikon *gear* maka akan keluar tampilan seperti gambar 6.9.



Gambar 6.9. Tampilan *Menu*

Menu ini juga sebenarnya adalah setting untuk mengatur semua yang akan digunakan dalam *game* seperti karakter, kelas atau kemampuan masing-masing karakter, dan kita juga bisa mengkustomisasi musuh di *menu* ini. Untuk di *game* saya sendiri, saya menggunakan karakter default tetapi diubah sedikit penampilannya supaya berbeda dari karakter lainnya. Saya juga menggunakan *resource* dari RPG Maker untuk tampilan dan *stats* dari musuh.



Gambar 6.10. Keterangan Musuh

Setelah kita membuat karakter, kita juga akan membuat peta yang akan digunakan pada saat *game* berlangsung. Membuat peta pun bisa dikustomisasi sendiri maupun menggunakan yang sudah disediakan oleh RPG Maker MV. Untuk membuat peta sendiri, kita dapat menggunakan elemen-elemen yang disediakan di bagian atas kiri seperti yang ada di gambar 6.8. Jika kita tidak ingin ribet atau memikirkan desain peta, di RPG Maker pun sudah disediakan peta yang langsung dapat kita gunakan untuk mengatur atau menaruh karakter dan musuh-musuh dalam *game* yang akan dibuat. Kita bisa mengatur peta dengan menggunakan *menu* peta yang berada di bawah elemen peta. Untuk melihat peta yang sudah disediakan, hanya cukup klik kanan pada bagian *menu* petanya.



Gambar 6.11. Pilihan Peta

Akan muncul beberapa pilihan seperti pada gambar 6.11, kita tinggal pilih saja peta yang akan digunakan dalam *game*. Jika butuh peta lagi, kita tinggal mengikuti langkah yang sama yaitu klik kanan pada *menu* peta dan pilihan peta pun akan keluar

kembali. Di game ini pun saya menggunakan peta yang sudah disediakan dari RPG Maker karena ini sangat menghemat waktu. Saya menggunakan 4 peta yang saya rancang untuk cerita dari *game* nantinya.



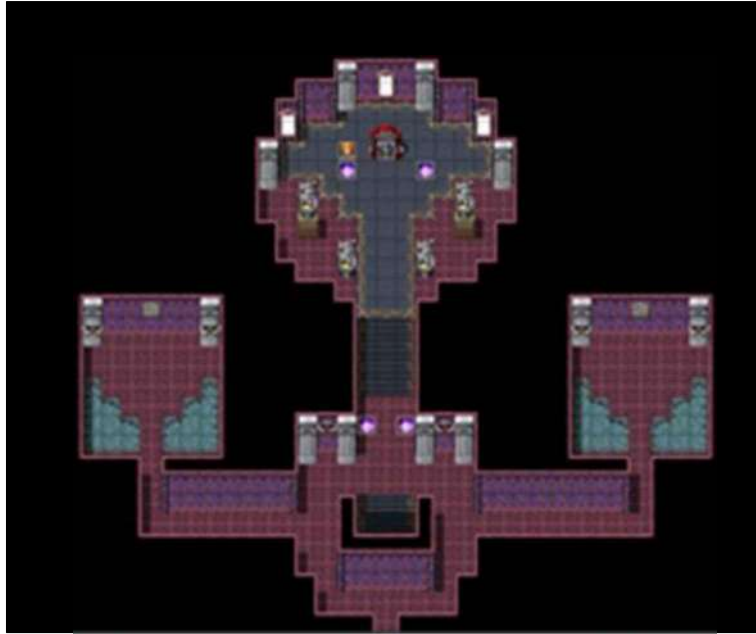
Gambar 6.12. Desa yang belum diketahui namanya



Gambar 6.13. Hutan menuju Kerajaan Kegelapan



Gambar 6.14. Bagian depan kerajaan kegelapan



Gambar 6.15. Bagian dalam kerajaan kegelapan

Untuk peta sendiri, saya merancang agar level di game ini mudah untuk dimainkan dan tidak perlu susah-susah untuk melanjutkan ke *level* selanjutnya.

NPC (Non-Playable Character)

Disini saya juga menggunakan *NPC (Non-Playable Character)* untuk saya gunakan sebagai warga yang akan memberikan pertanyaan supaya pemain bisa mendapatkan informasi mengenai jalan ke Kerajaan Kegelapan. Untuk *spawn NPC* pun caranya sangat mudah, misalkan kita klik pada kotak tertentu maka akan keluar kotak putih seperti *border* yang menandakan bahwa kita memilih kotak itu untuk diisi sesuatu.



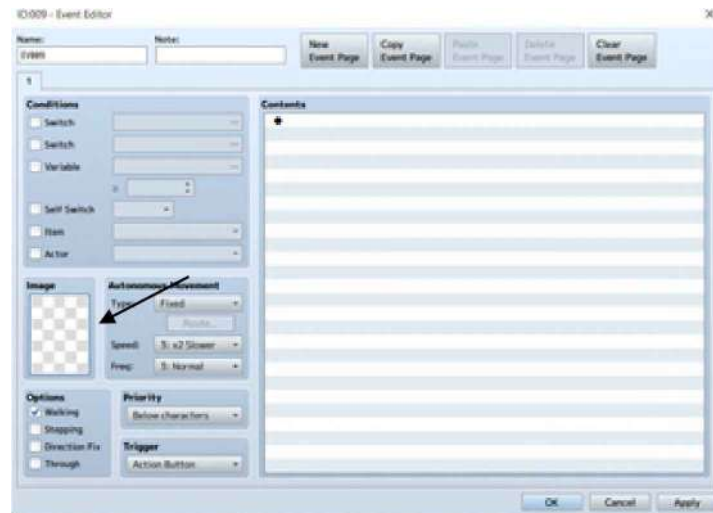
Gambar 6.16. Kotak dengan *border* putih

Setelah klik pada satu kotak, kita bisa klik kanan untuk menampilkan *menu* seperti pada gambar 6.16. Jika kita ingin membuat atau memunculkan *NPC* baru, kita klik *New* lalu kita pilih model atau tampilan *NPC*-nya.



Gambar 6.17. *Menu* yang keluar saat klik kanan

Jika kita sudah klik pada bagian *New*, maka akan keluar *menu*.
Disini kita bisa mengkustomisasi *NPC* hingga *event-event* atau *movement* dari *NPC* yang biasanya dibuat untuk *cutscene*.



Gambar 6.18. *Menu Event Editor*

Setelah keluar menu seperti gambar diatas, kita klik pada bagian yang sudah ditandai anak panah pada gambar 6.18. Itu adalah menu untuk memilih tampilan *NPC* yang akan kita gunakan sebagai sumber informasi.



Gambar 6.19. Menu tampilan *NPC*

Pada gambar diatas adalah contoh menu *NPC* yang bisa kita pilih sesuka hati kita dengan klik salah satu posisi dari *NPC* yang akan kita gunakan.

Untuk contoh *NPC* yang akan saya gunakan dalam *game* terdapat dalam gambar 6.20.



Gambar 6.20. Contoh *NPC*

Jika sudah, kita juga bisa mengkustomisasi misal dialog atau apa saja yang akan dilakukan oleh *NPC* yang sudah kita buat. Saya akan berikan contoh saat karakter kita memulai percakapan dengan *NPC*. Contoh akan saya berikan pada gambar 6.21 dan 6.22.

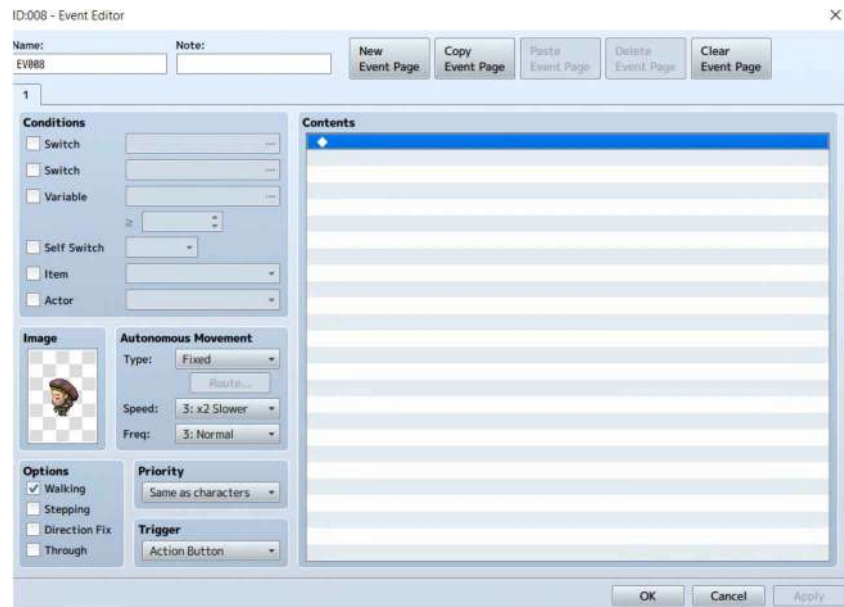


Gambar 6.21. Contoh dialog



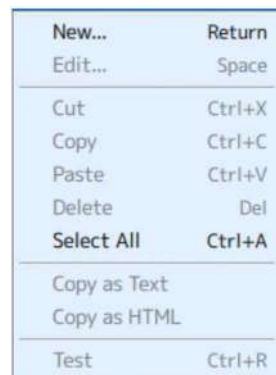
Gambar 6.22. Contoh dialog

Cara untuk membuat dialog pun sangat mudah, jika kita ingin memulai percakapan dari *NPC* terlebih dahulu maka kita klik kanan pada bagian *NPC* lalu akan keluar tampilan seperti pada gambar di bawah 6.23.



Gambar 6.23. Tampilan *Event Editor NPC*

Setelah itu kita klik pada salah satu baris sehingga berubah menjadi warna biru tua seperti pada gambar 6.23. Setelah itu, klik kanan pada baris yang sudah berubah menjadi warna biru tua itu dan akan keluar tampilan menu lagi.



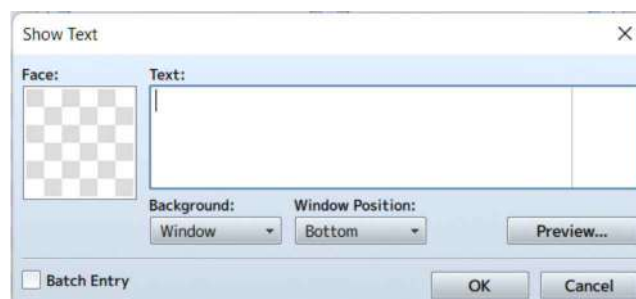
Gambar 6.24. Menu setelah klik kanan

Klik new dan akan keluar tampilan seperti pada gambar di bawah ini.



Gambar 6.25. Menu *event commands*

Untuk memunculkan dialog, klik pada *show text*. Setelah klik *show text* akan keluar menu lagi untuk menulis dialog yang ingin kita tuliskan dan pilih orang yang melakukan dialog tersebut.



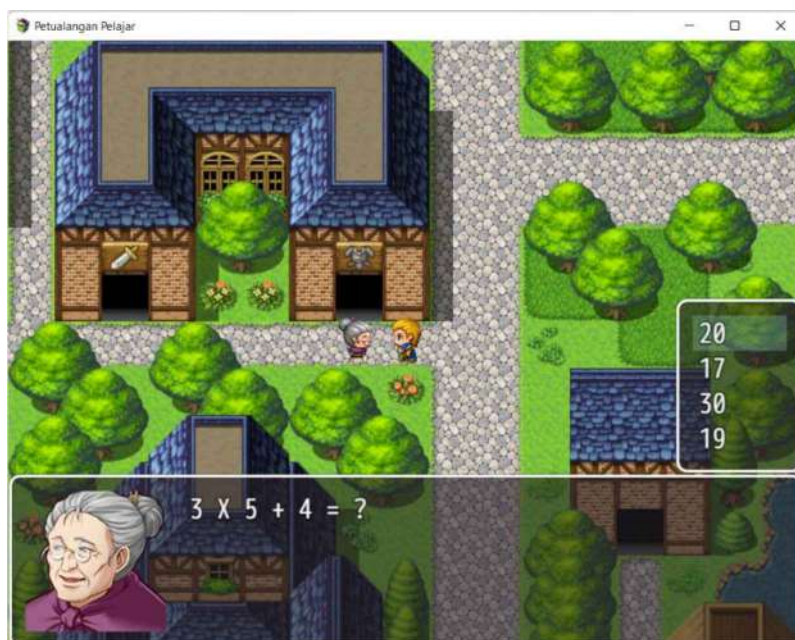
Gambar 6.26. Menu *show text*

Jika sudah, kita tinggal pilih karakter pada kolom *face* tuliskan dialognya di kolom *text*. Lalu klik *ok* untuk selesai.

Contoh Pertanyaan

Karena *game* ini saya buat dengan adanya edukasi, maka saya juga menambahkan pertanyaan yang berdasarkan pelajaran yang mudah dimengerti oleh anak-anak seperti matematika dan pengetahuan umum. Pertanyaan di *game* ini saya buat supaya pemain mendapat informasi dari para warga yang ada di desa.

Lalu untuk contoh pertanyaan yang akan ditanyakan kepada pemain supaya mendapatkan informasi dari warga akan saya tunjukkan pada gambar 6.27.



Gambar 6.27. Contoh pertanyaan

Disini pemain harus memilih jawaban yang benar. Jika salah dalam memilih maka pemain dapat mengulanginya kembali dan jika benar maka akan diberitahu jalan menuju Kerajaan Kegelapan.



Gambar 6.28. Tampilan jika salah



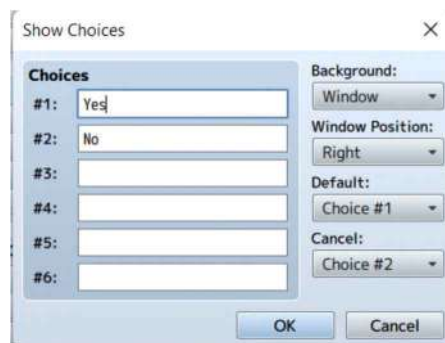
Gambar 6.29. Tampilan jika benar

Untuk membuat pilihan pun hampir sama dengan membuat dialog, tetapi disini kita klik *show choices*.



Gambar 6.30. Menu *event commands*

Setelah klik *show choices*, akan keluar menu lagi dengan tampilan seperti di bawah ini.



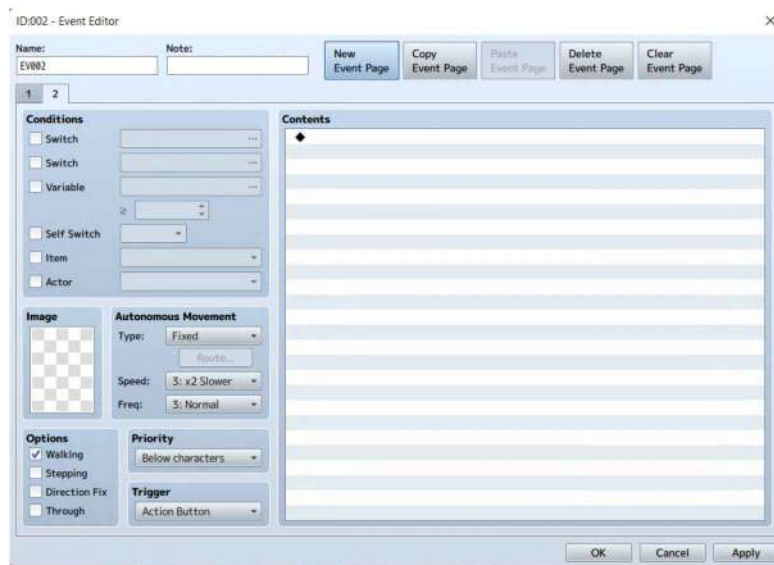
Gambar 6.31. Menu *show choices*

Disini kita bisa menulis pilihan jawaban yang akan diberikan ke pemain untuk dijawab.

Teleport

Untuk *teleport* ini saya gunakan untuk karakter kita di game sehingga dapat berpindah dari peta pertama ke peta berikutnya. Cara untuk menggunakan *teleport* ini adalah

dengan klik kanan pada salah satu kotak pada peta lalu klik *New* hingga muncul menu seperti gambar berikut.



Gambar 6.32. Menu *Event Editor*

Lalu klik sekali pada salah satu baris sehingga menjadi biru seperti pada gambar di bawah ini.



Gambar 6.33. Menu *Event Editor* setelah di klik pada salah satu barisnya

Klik kanan hingga keluar menu kecil lalu klik *New*. Setelah itu akan keluar *event commands* yang ada berbagai macam *event* untuk dipilih sesuai dengan kebutuhan kita. Disini saya akan mengajarkan cara untuk memberikan *teleport* karakter kita. Untuk *teleport* sendiri, kita akan menggunakan *transfer player* yang ada di halaman 2 *event commands*.



Gambar 6.34. Menu *Event Commands*

Jika sudah klik pada *transfer player* maka akan keluar menu lagi dengan tampilan seperti pada gambar 4.28.



Gambar 6.35. Menu *Transfer Player*

Pada menu ini kita dapat mengatur dari mana karakter kita akan berpindah dan menentukan tujuannya. Jika kita klik pada

Forest Town (0,0) maka akan keluar lagi tampilan peta untuk meletakkan posisi karakter kita akan berpindah dengan menandai satu kotak yang nantinya ujung kotak itu akan ada garis putih di sekitarnya. Itu adalah tempat dimana karakter kita berpindah nantinya.



Gambar 6.36. Pemilihan lokasi *spawn*

Jika sudah berhasil, seharusnya karakter kita akan berpindah ke peta berikutnya.

Dialog jika pemain menang atau kalah dalam bertarung

Dalam suatu *game* pastinya jika menang atau kalah tidak mungkin tidak ada lanjutan atau misalkan dialog yang akan keluar jika menang atau kalah karena kita tidak akan tau lanjutan dari pertarungan itu jika *game* yang kita mainkan tidak jelas atau akan berakhir sama saat menang maupun kalah. Maka dalam *game* ini saya buat juga dialog saat pemain menang atau kalah karena akan sangat berpengaruh terhadap kelanjutan *gamenya*.



Gambar 6.37. Tampilan jika menang



Gambar 6.38. Tampilan jika menang



Gambar 6.39. Tampilan jika kalah



Gambar 6.40. Tampilan jika kalah



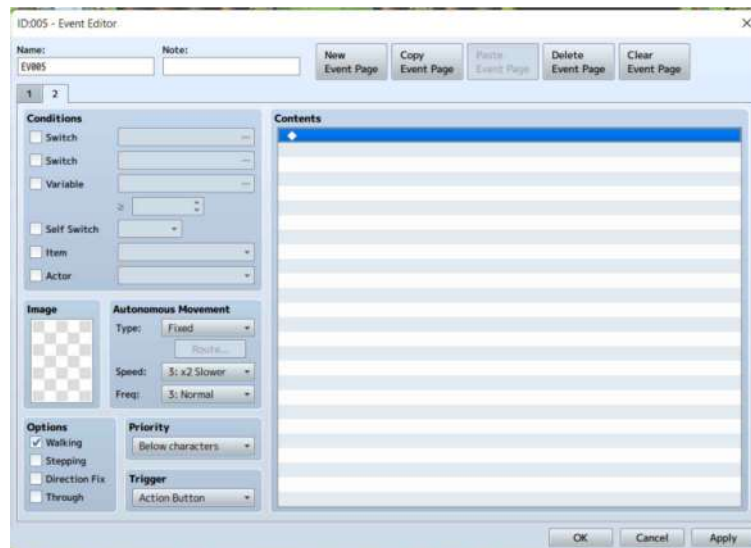
Gambar 6.41. Tampilan jika kabur



Gambar 6.42. Tampilan jika kabur

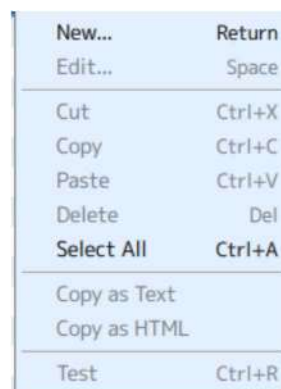
Untuk dialog yang akan keluar setiap menang atau kalah akan berbeda-beda, jadi tidak semua sama seperti pada gambar diatas.

Untuk membuat tampilan seperti di atas, kita hanya perlu *spawn* musuh lalu kita klik kanan setelah itu klik *new*.



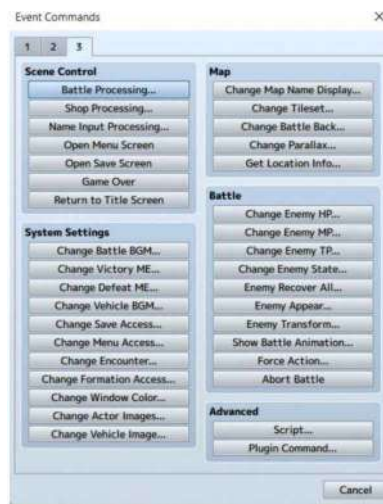
Gambar 6.43. Menu *event editor* musuh

Lalu klik salah satu baris hingga berubah menjadi warna biru tua seperti pada gambar, sesudah itu klik kanan hingga keluar menu seperti pada gambar di bawah.



Gambar 6.44. Menu setelah klik kanan

Akan keluar menu lagi setelah kita klik *new*, klik pada nomor 3 pada bagian atas.



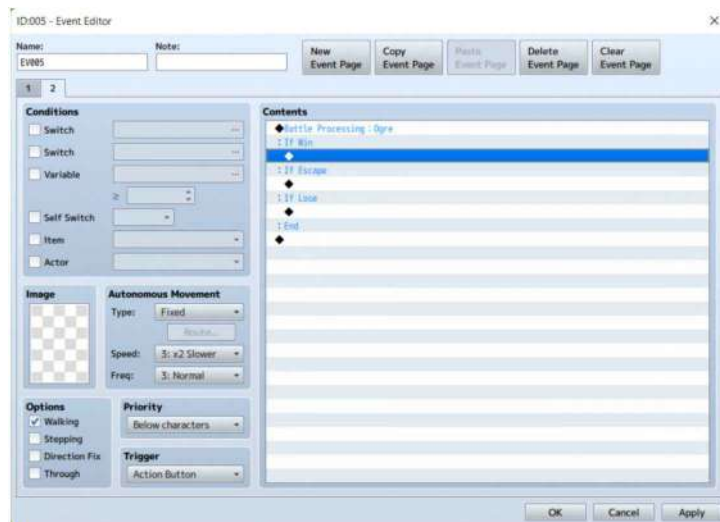
Gambar 6.45. Menu *event commands*

Setelah menu ini keluar, klik pada bagian *battle processing* untuk memberi *event* baru yaitu bertarung. Akan muncul menu lagi untuk memilih musuh dan memilih bisa kabur ataupun kalah.



Gambar 6.46. Menu *battle processing*

Jika sudah klik *ok*, maka akan keluar seperti isian kosong yang dapat diisi dengan dialog dari musuh semisal kalah atau menang.

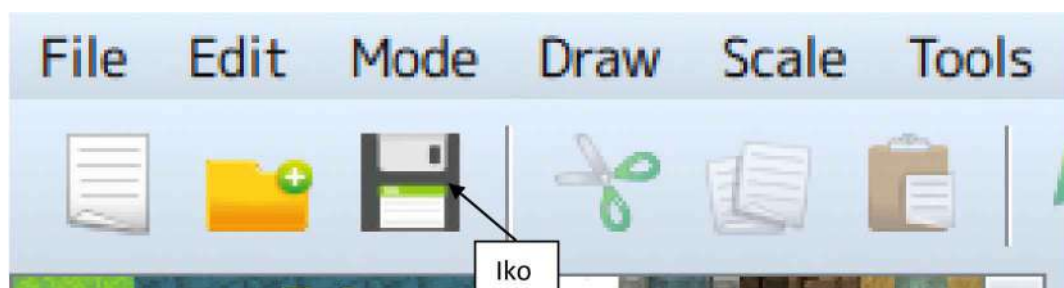


Gambar 6.47. Isian yang dapat diisi untuk dialog musuh

Cara ekspor game

Jika kita sudah selesai dalam membuat *gamenya* maka *game* yang akan kita buat akan kita *save* supaya semua progress dapat tersimpan dengan baik. Setelah *save* pun jika kita ingin memainkan hasil jadinya, maka kita dapat menggunakan ekspor untuk ekspor hasil jadi *gamenya*.

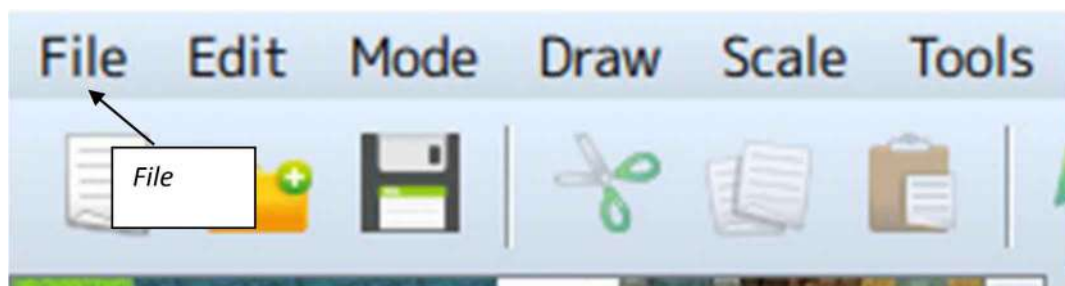
Cara untuk *save* progress *game* akan saya tunjukkan pada gambar di bawah ini.



Gambar 6.48. Ikon Save pada bagian atas RPG Maker MV

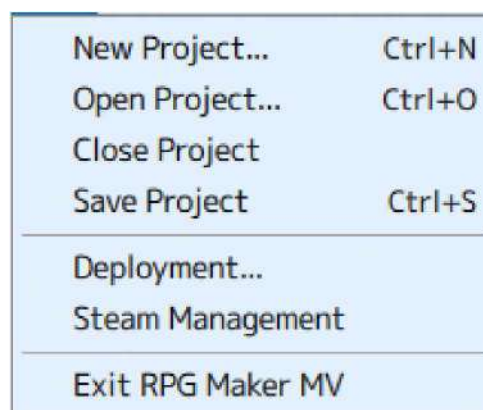
Untuk *save*, kita hanya perlu klik pada ikon *save* seperti pada gambar 4.41.

Dan untuk ekspor sendiri caranya dengan klik pada *File* pada bagian atas ikon *New Project*.



Gambar 6.49. Keterangan *File*

Lalu jika sudah klik maka akan keluar menu yang akan saya tampilkan di gambar selanjutnya.



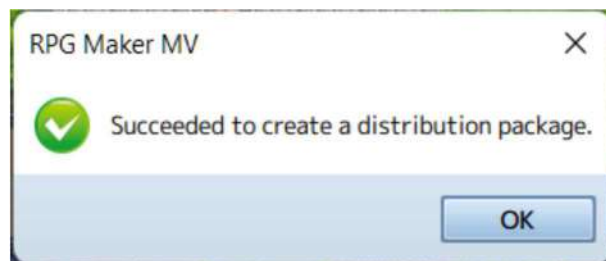
Gambar 6.50. Menu setelah klik pada *File*

Untuk ekspor, kita klik lagi pada *Deployment*. Menu akan keluar lagi setelah klik *Deployment*. Menu ini adalah untuk mengatur mulai dari *platform*, *options*, *encryption*, dan lokasi untuk menyimpan *project game* kita.



Gambar 6.51. Menu *Deployment*

Jika sudah selesai mengatur semuanya, kita klik pada *ok* untuk menyelesaikan *game* kita.



Gambar 6.52. Tampilan jika ekspor berhasil

Jika berhasil dalam ekspor, akan muncul tampilan seperti pada gambar 6.52.

Link gamenya akan saya share di bawah ini:

Link *game*:

https://drive.google.com/file/d/1HOaISVr6nK_69kvgjiJ-LwhOeBis-FSv/view?usp=sharing

Dragon Hunter

Renaldo Haryono Kusumo

Bermain adalah kegiatan yang dilakukan oleh semua anak, dan bahkan dikatakan bahwa anak-anak menghabiskan waktunya untuk bermain *game*. Kamus besar Bahasa Indonesia (2003: 697) menyatakan bahwa bermain berarti melakukan sesuatu dengan kesenangan (dengan atau tanpa alat tertentu). Permainan di picu, berkembang, dan tumbuh dengan kekuatan yang tersisa. Pembangkitan kekuatan dalam diri seorang anak melebihi apa yang diperlukan secara fisik maupun psikologis.

Kontribusi industri kreatif juga memiliki peringkat terhadap Produk Domestik Bruto (PDB) nasional yang mempunyai peningkatan yang linear atau semakin meningkat tiap tahunnya, dari situ juga kita bisa melihat bahwa *game* semakin digemari oleh banyak orang tanpa memandang umur, salah satu permainan yang menjadi unggulan cukup besar di PDB adalah permainan yang bertemakan tentang edukasi.

Bermain juga merupakan kegiatan dimana anak-anak berulang kali terlibat di dalam kesenangan tanpa harus mencapai tujuan mereka. Rata-rata usia anak mengenal *game* di antara rentan usia 4-5 tahun, dengan tahap pengenalan lingkungan sekitar, juga membantu mengaktifkan saraf motorik halus pada anak-anak seperti melalui *game puzzle*. Kegiatan bermain tidak hanya diminati oleh kalangan anak-anak saja, melainkan diminati oleh banyak kalangan termasuk

kalangan dewasa dan lansia dengan maksud dan tujuan yang berbeda-beda. Seperti halnya pada orang dewasa, kegiatan bermain ini membantu mereka merelaksasikan pikiran setelah dihadapi dengan berbagai pekerjaan. Juga terhadap kaum lansia, kegiatan bermain dapat membantu mereka mempertahankan sistem kerja otak mereka, salah satunya menghindari terkenanya *Alzheimer* pada kaum lansia.

Di dunia yang sudah maju ini, kegiatan bermain tidak hanya untuk bersenang-senang saja melainkan dimanfaatkan untuk memperluas jaringan pertemanan, terkhusus untuk kaum milenial. Di era modern ini, bermain suatu permainan (*game*) juga dimanfaatkan sebagai usaha untuk mendapatkan pemasukan (*income*). Banyak orang berlomba-lomba menciptakan suatu permainan ataupun meningkatkan kualitas permainan mereka untuk menjadi pemain yang baik (*pro player*) sehingga pendapatan pun semakin meningkat. Salah satu caranya dengan melakukan *live streaming* ketika sedang bermain *games* atau melakukan kegiatan promosi di berbagai *platform* seperti Youtube, Instagram, Tik-Tok, serta media sosial lainnya.

Tetapi di tahun 2022 ini kita dipersulit dengan datang nya wabah Bernama COVID-19 yang membuat rakyat Indonesia untuk melakukan semua kegiatan dari dalam rumah termasuk bekerja, dari wabah tersebut membuat jurusan yang berkaitan dengan dunia *online* semakin diminati banyak orang terutama dalam membuat *game*, tetapi dibalik itu masih banyak orang yang tidak mengerti dengan bahasa pemrograman atau *coding*, terutama dengan berbagai macam *game engine* yang

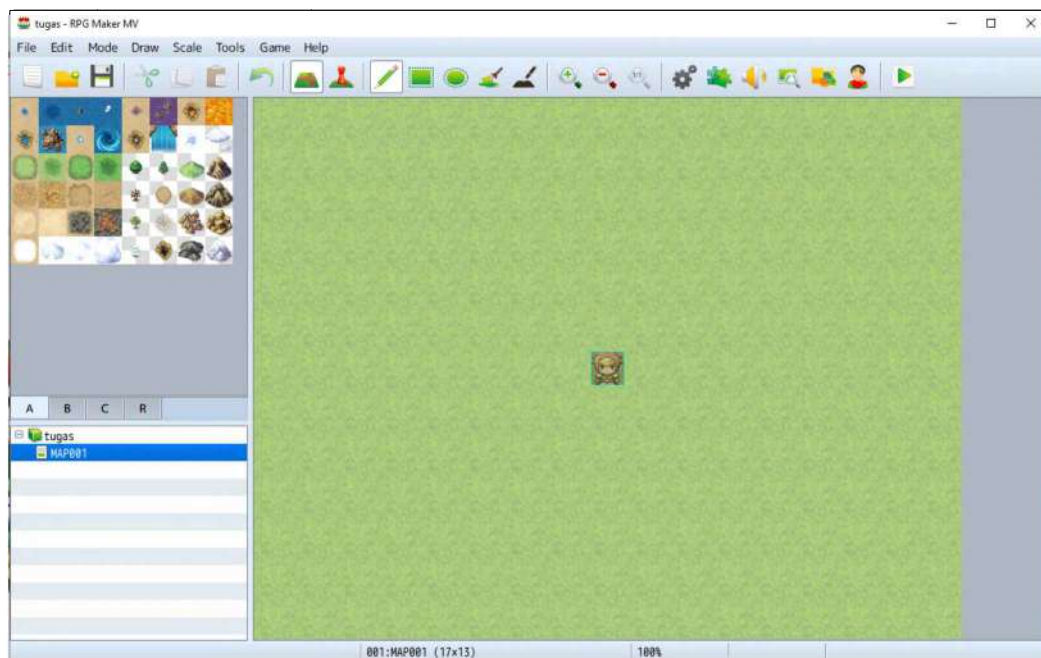
membutuhkan Bahasa pemrograman, padahal banyak *game engine* yang tidak harus memikirkan Bahasa pemrograman tersebut, contohnya seperti RPG Maker MV, di dalam *game engine* tersebut kita tidak perlu memikirkan tentang *coding* karena di RPG Maker MV sudah disediakan semua logika-logika dalam pembuatan *game*.



Gambar 7.1. Lambang RPG Maker MV

aplikasi *game engine* ini dapat dioperasikan di berbagai *platform* seperti Windows, dan iOS. Aplikasi ini bisa dengan mudah untuk dikuasai karena kita dapat membuat *game* yang kita mau tanpa ada nya *coding* atau bahasa pemrograman.

RPG Maker MV sendiri adalah aplikasi pembuat *game* yang ber-genre *role-playing video games*, di aplikasi ini sendiri kalian bisa membuat imajinasi kalian menjadi nyata karena di aplikasi *game engine* ini kalian sudah dipermudah dengan adanya alat-alat seperti tanah, tembok, air, pegunungan, dan masih banyak lagi.



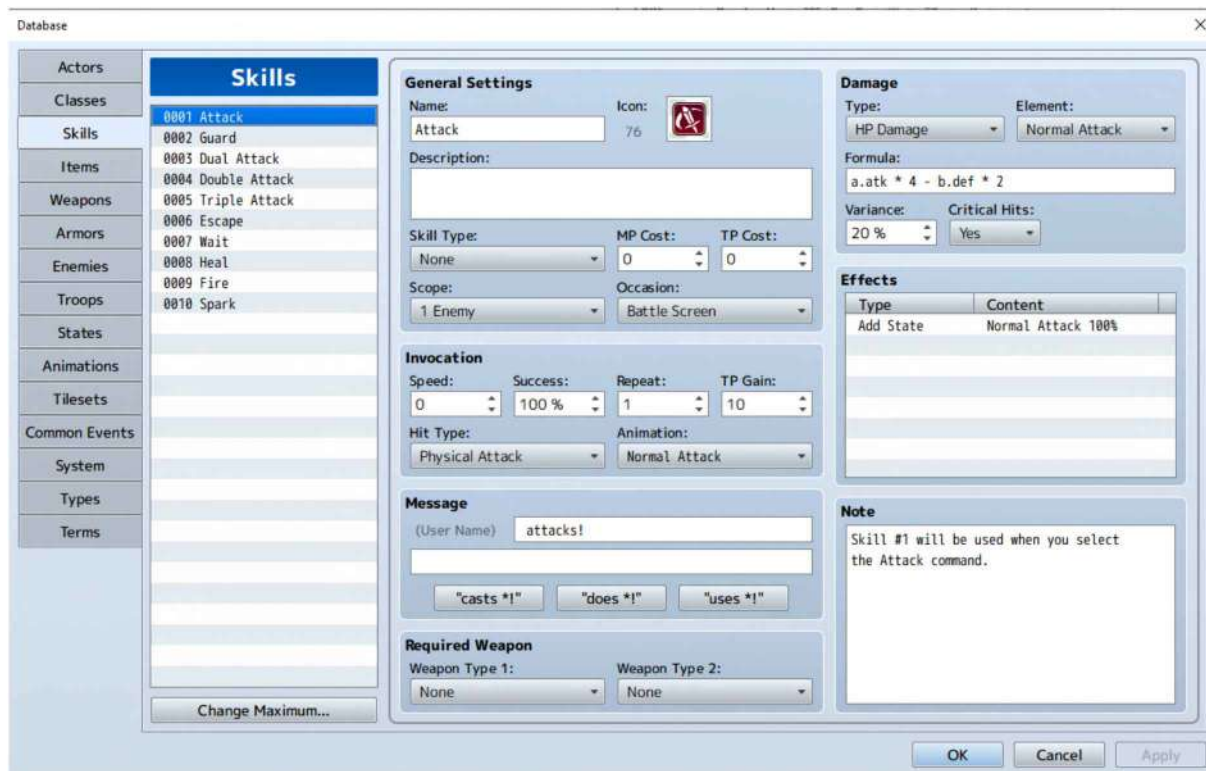
Gambar 7.2. Tampilan RPG Maker MV

Ketika kalian mengunduh aplikasi RPG Maker MV kalian akan disuguhi beberapa macam pilihan untuk membuat *game*, ketika tampilan seperti contoh pada gambar 1.2. muncul maka di situ kalian sudah bisa membuat *game* yang kalian inginkan sesuai dengan imajinasi kalian.



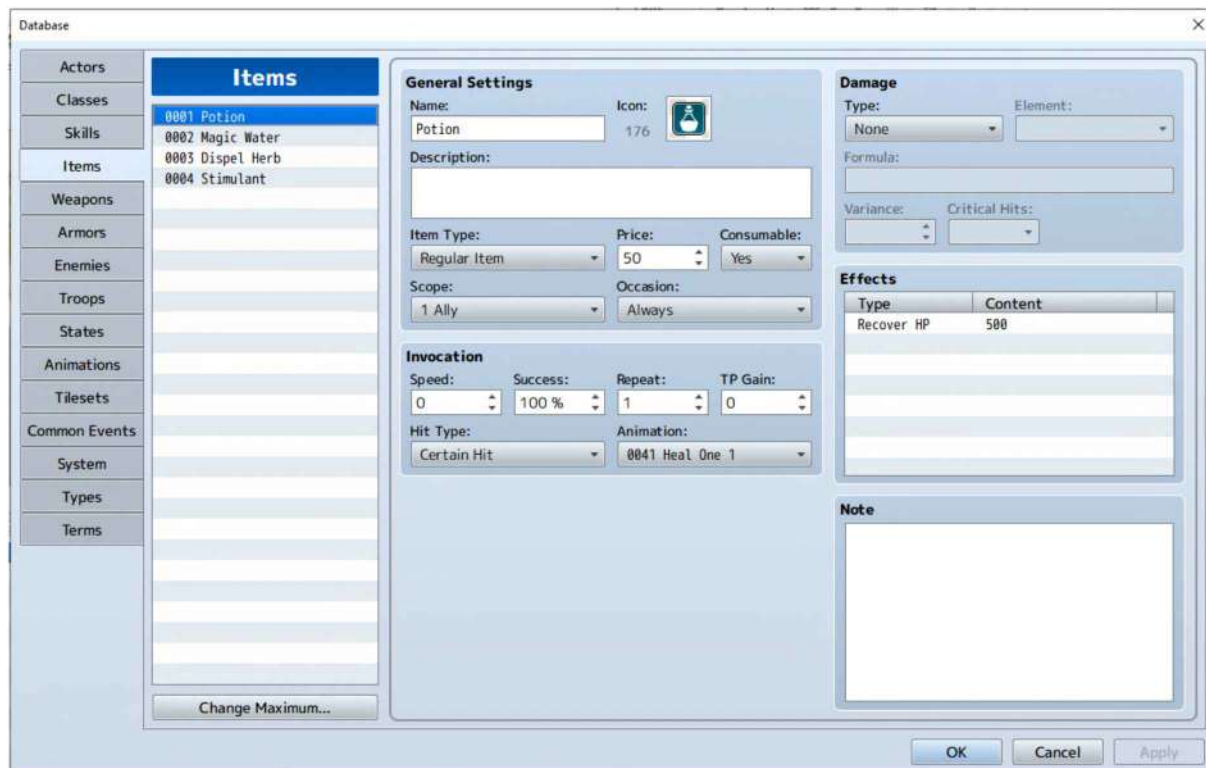
Gambar 7.3. Tampilan karakter di dalam *game*

Di pilihan ini kalian bisa menambahkan karakter pemain yang kalian inginkan, ketika kalian bosan dengan karakter yang begitu saja, kalian bisa membuat karakter kalian sendiri di dalam aplikasi atau kalian juga bisa mencari karakter sesuai kesukaan kalian di aplikasi pencari seperti Google dan lainnya.



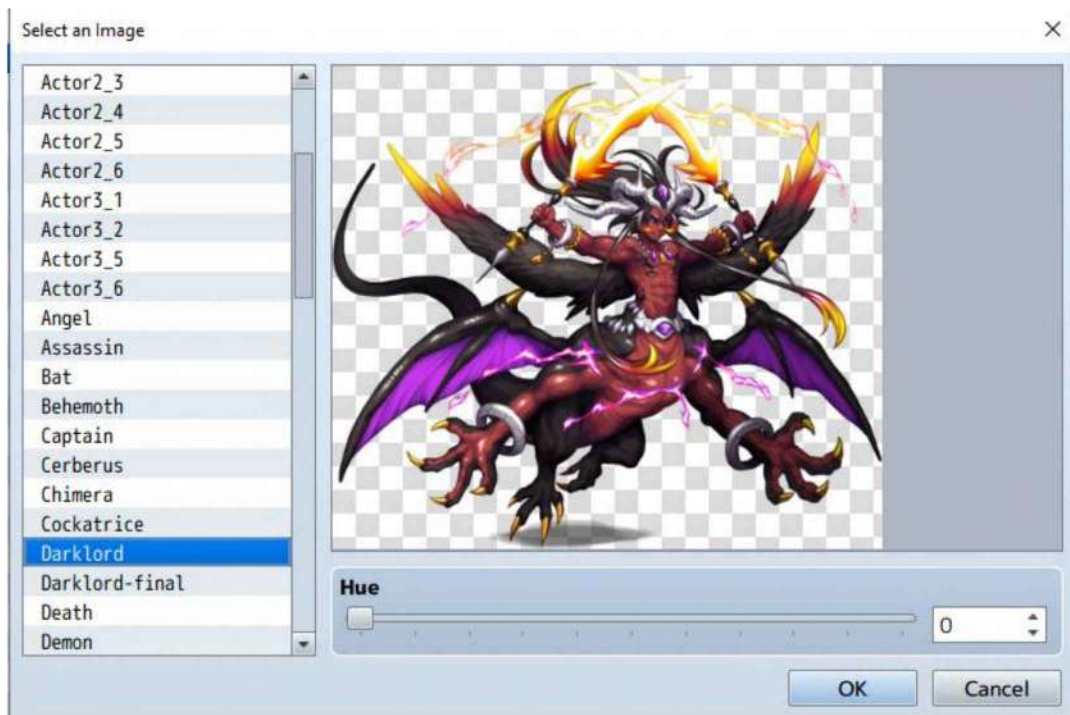
Gambar 7.4. Tampilan kemampuan di dalam *game*

Di bagian pilihan ini kalian di suguhi beberapa macam kemampuan yang kalian inginkan di karakter kalian masing-masing.



Gambar 7.5. Tampilan peralatan di dalam *game*

Di pilihan ini kalian akan disuguhi berbagai macam peralatan seperti pedang, kapak, perisai, dan lain-lain untuk melawan *monster-monster* di dalam *game*.



Gambar 7.6. Tampilan karakter musuh di dalam *game*

Di bagian pilihan ini kali bisa memilih lawan kalian yang akan kalian lawan di dalam *game* yang akan kalian buat, bukan hanya memilih tapi kalian juga bisa menyesuaikan kekuatan para *monster* agar pemainnya bisa memiliki tingkat kesulitan sesuai dengan yang kalian mau.

Apa sih artinya *RPG* atau *role-playing video games*? *RPG* atau *role-playing video games* adalah *game* yang berfokus pada karakter yang kita tentukan sendiri, contohnya kamu adalah seorang raja yang sangat terkenal dengan keadilan yang kamu buat, lalu kamu mendapatkan kabar tentang kejahatan dan kamu yang harus menyelesaikannya, Maka di *scene* tersebut kamu akan di buat berfikir untuk menyelesaikan

permasalahan tersebut secara adil tanpa adanya membebani pihak lain.

Nah kelebihan dari *game* ber-genre *RPG* atau *role-playing video games* ini adalah :

1. *Game RPG* dinilai mampu memberikan banyak manfaat dalam tumbuh kembang seorang anak.
2. Anak yang memainkan *game* ber-genre *RPG* dinilai mempunyai cara bersikap(*manner*) yang lebih baik daripada anak seusianya. Hal tersebut disebabkan dari dalam *game RPG* mempunyai jalan cerita yang melihatkan etika-etika bersosialisasi dengan baik.
3. Seorang anak dapat mengartikulasikan emosinya secara *responsive* di dunia nyata.
4. Logika berfikir anak yang diintegrasikan dengan berbagai pilihan dalam *game RPG* dapat membuat peningkatan pengalaman dalam keterampilan kognitif seorang

pemain dalam berkomunikasi dan menentukan alur pilihan.

5. Bahkan *game RPG* dinilai mampu digunakan sebagai alat penyembuhan bagi orang yang mengalami tekanan psikologis.

TANTANGAN GAME

Di dalam *game* yang dibuat oleh RPG Maker MV ini kalian akan disuguhi dengan berbagai macam teka-teki, dan segala macam gangguan dari musuh-musuh dari yang terlemah hingga yang terkuat.

Di dalam *game* ini tentunya kalian akan dibantu jika kalian dapat mengalahkan segala musuh yang kalian lawan, karena setiap kali kalian membunuh *monster-monster* di masing-masing tempat kalian akan mendapatkan barang-barang yang bisa membantu kalian untuk membunuh raja terkuat di dalam *game* tersebut.

Game ini menceritakan tentang petarung yang di panggil oleh raja setempat untuk membasmi naga yang sangat mengganggu para penduduk di kerajaan tersebut.

Mekanisme Game

Pemain akan menjalankan karakter utama, pemain akan diberikan tugas oleh raja dari suatu wilayah untuk memusnahkan naga yang sangat mengganggu rakyat setempat, tentunya pemain tidak bisa langsung melawan musuh terakhir karena perbedaan kekuatan yang cukup jauh, maka diadakan nya musuh-musuh kecil yang akan memberikan pemain sebuah barang-barang berguna untuk membantu membasmi naga tersebut.

Game

Game “Dragon Hunter”, adalah sebuah game yang ber-genre role-playing video games. Game ini ditujukan untuk pemain berusia 13 tahun keatas.

Karakter Dalam Game

Di dalam *game* ini tentunya memiliki beberapa macam karakter. Untuk karakter utama bernama Alex, dan beberapa *NPC* seperti rakyat dan raja.

Karakter utama :



Gambar 7.7. Alex

Karakter kerajaan :



Gambar 7.8. Raja



Gambar 7.9. Ratu



Gambar 7.10. Panglima Kerajaan



Gambar 7.11. : Penasihat Kerajaan

Karakter rakyat :



Gambar 7.12. *NPC*

Karakter musuh :



Gambar 7.13. Naga

TAHAPAN MEMBUAT GAME

Sebelum kalian membuat *game* di RPG Maker MV kalian harus mengetahui 2 fitur yang penting di dalam pembuatan *game RPG*, yaitu *event* dan *map*.

Event

Event adalah tempat dimana peristiwa di dalam permainan itu akan terjadi, seperti melakukan perpindahan tempat menjawab teka-teki, dan lain-lain.

Tampilan *event* :



Gambar 7.14. Pertempuran melawan naga

Map

Map atau peta adalah tempat untuk membuat gambaran dalam permainan seperti membuat bangunan, alam sekitar barang-barang pelengkap, dan lain-lain

Tampilan *map* :



Gambar 7.15. Tampilan *map*

Link men-*download game* :

https://drive.google.com/file/d/1-05fA-QnHoVas91IPMD-TrGbKw_jrJma/view?usp=sharing

Musik Mini Quiz

Justine Hezekiel Ramli

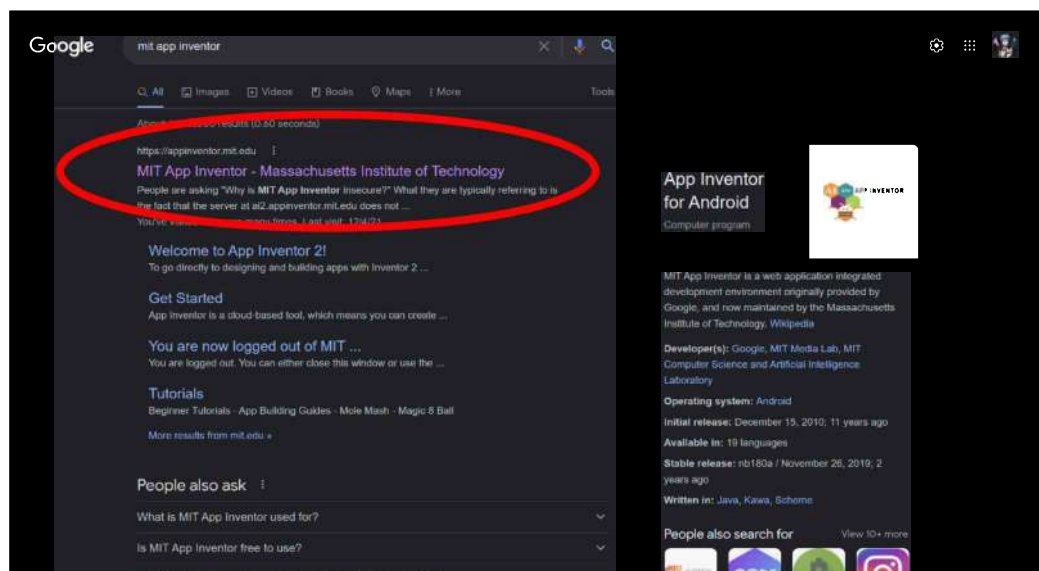
Jaman sekarang *game* merupakan salah satu faktor penting dalam kehidupan beriringan dengan kemajuan teknologi. Sekarang *game* menjadi sesuatu yang hidup disekitar masyarakat berbeda dengan beberapa tahun yang lalu dimana stigma masyarakat terhadap *game* masih sangat buruk, karena hanya buang buang waktu atau dapat memberikan efek negatif bagi yang memainkan. Namun semakin berkembangnya jaman *game* sekarang sudah mulai mendapat pandangan yang cukup baik karena dapat meredakan stres ataupun banyak orang yang dapat penghasilan dari *game*.

Kemudian sekarang *game* juga memiliki berbagai genre seperti *Arcade*, *FPS*, *MOBA*(*Multiplayer Online Battle Arena*), *Race*, *Quiz*, *puzzle*, *Detective*, *RPG*, Musik, *Strategy*, Simulasi, Edukasi, dan lain-lain. Dari masing masing genre mempunyai keseruan yang berbeda, seperti *game-game Quiz*, *Puzzle*, dan Detektif yang membutuhkan kemampuan kognitif. Kemudian *game-game* bergenre *MOBA* dan *FPS* yang membutuhkan kerja sama tim. Kemudian *game-game* Simulasi, *Arcade* atau *RPG* yang membutuhkan kesabaran.

Sekarang juga banyak *game* berbasis musik atau simulasi alat musik yang dapat kita gunakan untuk mengasah bakat atau kemampuan pemain di dalam bidang musik. Dengan adanya simulasi alat musik kita dapat bermain alat musik dimana saja kita mau. Dalam tulisan pertamaku ini aku ingin menceritakan

proses pembuatan *game* yang berhubungan dengan musik yang merupakan *hobby*-ku. *Game* yang akan aku buat kali ini akan menggunakan MIT ataupun RPG Maker yang dimana tidak membutuhkan *coding* dalam pembuatannya.

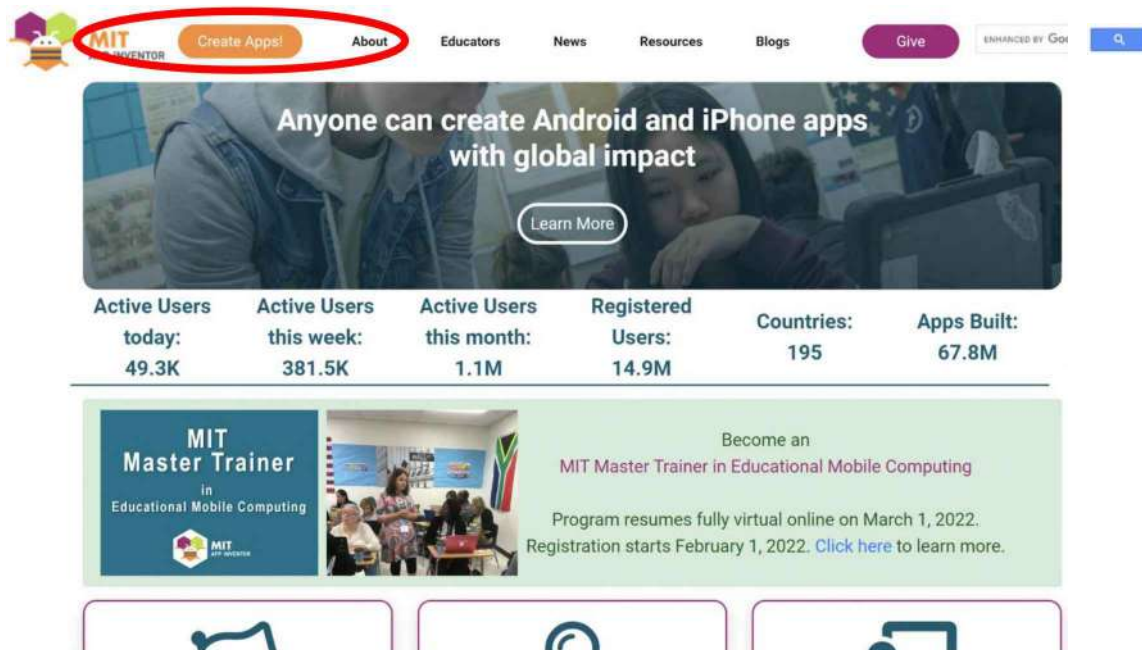
Game yang akan aku buat bergenre *Quiz* dengan tema musik. Dan aplikasi yang aku pilih untuk membuat *game* adalah MIT App Inventor. MIT App Inventor adalah aplikasi *web open source* yang dikembangkan oleh Google yang sekarang dikelola oleh Massachusetts Institute of Technology. MIT ditujukan untuk para pemula yang ingin membuat aplikasi perangkat lunak dengan basis sistem Android.



Gambar 8.1. Website appinventor.mit.edu

Aku sendiri awalnya mengenal MIT App Inventor dari materi yang aku dapat di kuliah semester 1. Kita dapat mengakses MIT App Inventor *via browser* dengan mengetikkan kata kunci MIT App Inventor

Kemudian kita akan diarahkan ke halaman beranda dari MIT App Inventor disitu kita dapat menekan *“Create Apps!”* untuk memulai



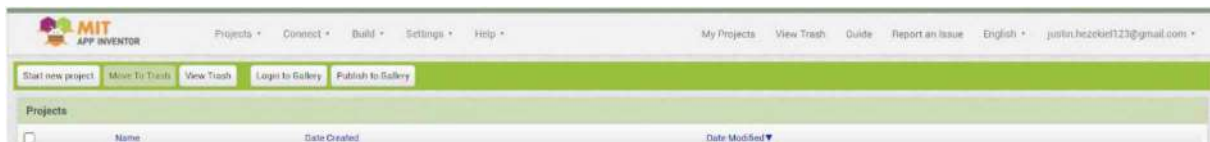
Gambar 8.2. Website appinventor.mit.edu

Setelah itu kita akan diarahkan ke halaman pemilihan akun, setelah memilih akun kita akan masuk ke tampilan awal sebelum membuat aplikasi.



[Privacy Policy and Terms of Use](#)

Gambar 8.3. Website appinventor.mit.edu



[Privacy Policy and Terms of Use](#)

Gambar 8.4. Website appinventor.mit.edu

Setelah itu kita dapat menekan “*Start New Project*” kemudian akan diminta menamai *project* yang akan kita buat, setelah itu tekan OK.



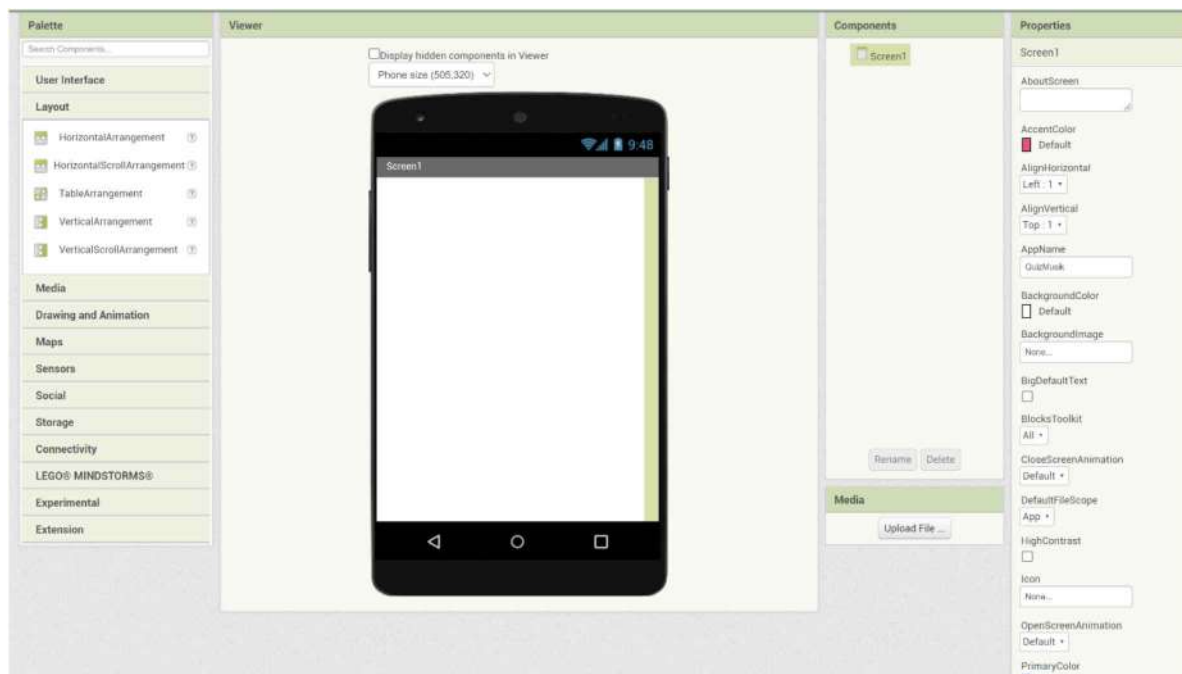
Create new App Inventor project

Project name:

Cancel OK

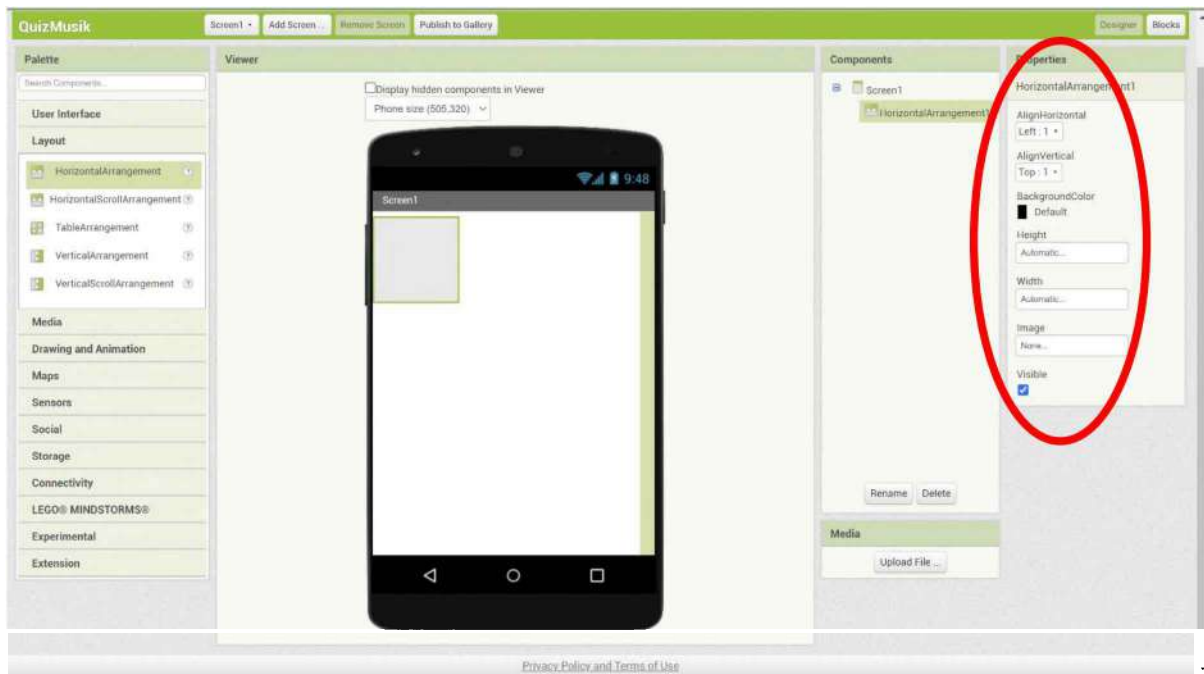
[Privacy Policy and Terms of Use](#)

Gambar 8.5. Memberi judul *project*



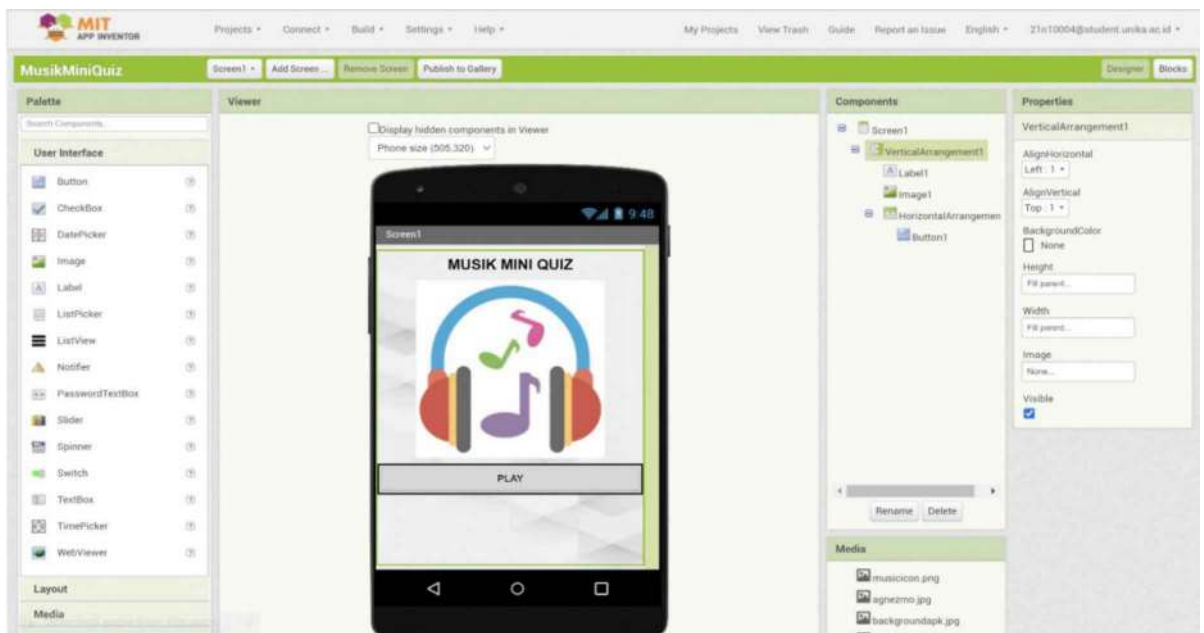
Gambar 8.6. Dashboard website appinventor.mit.edu

Kita bisa memulai membuat aplikasi yang kita ingin dengan memasukkan *layout* yang kita pilih diantaranya ada, *horizontal Arrangement*, *Horizontal Scroll Arrangement*, *Table Arrangement*, *Vertical Arrangement*, *Vertical Scroll Arrangement*.



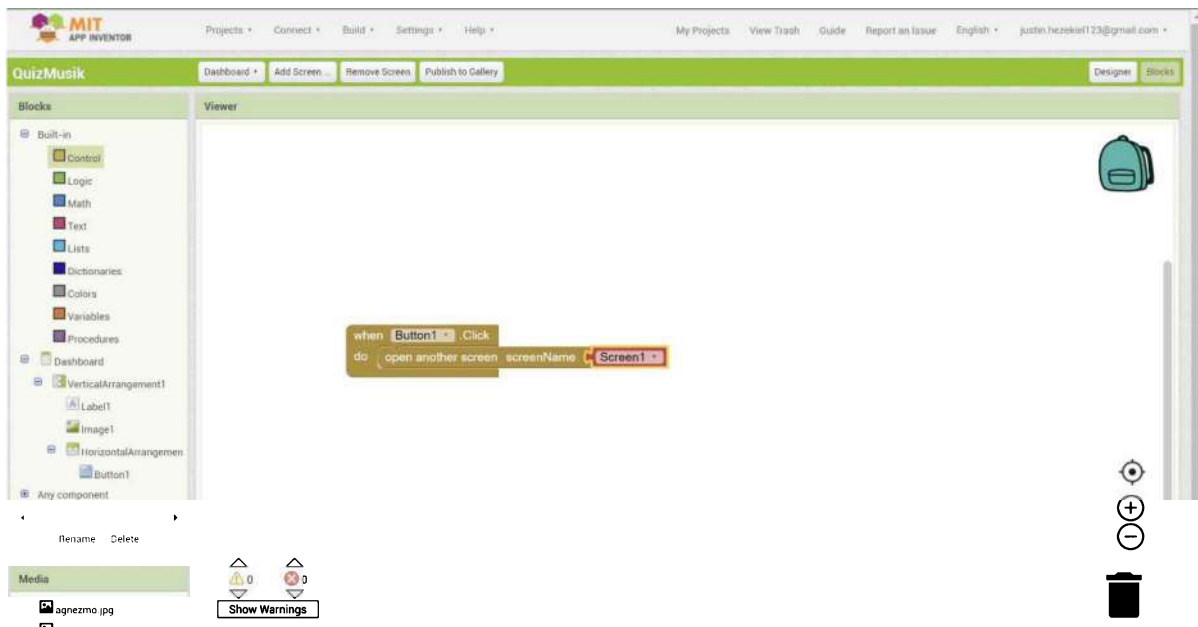
Gambar 8.7. Pengenalan *tool*

Kemudian dibagian kanan kita dapat mengatur seputar *layout* kita. Dari *Align Horizontal*, *Align Vertical*, *Background Color*, *Height*, *Width*, dan *Image*.



Gambar 8.8. Tampilan *dashboard screen* aplikasi

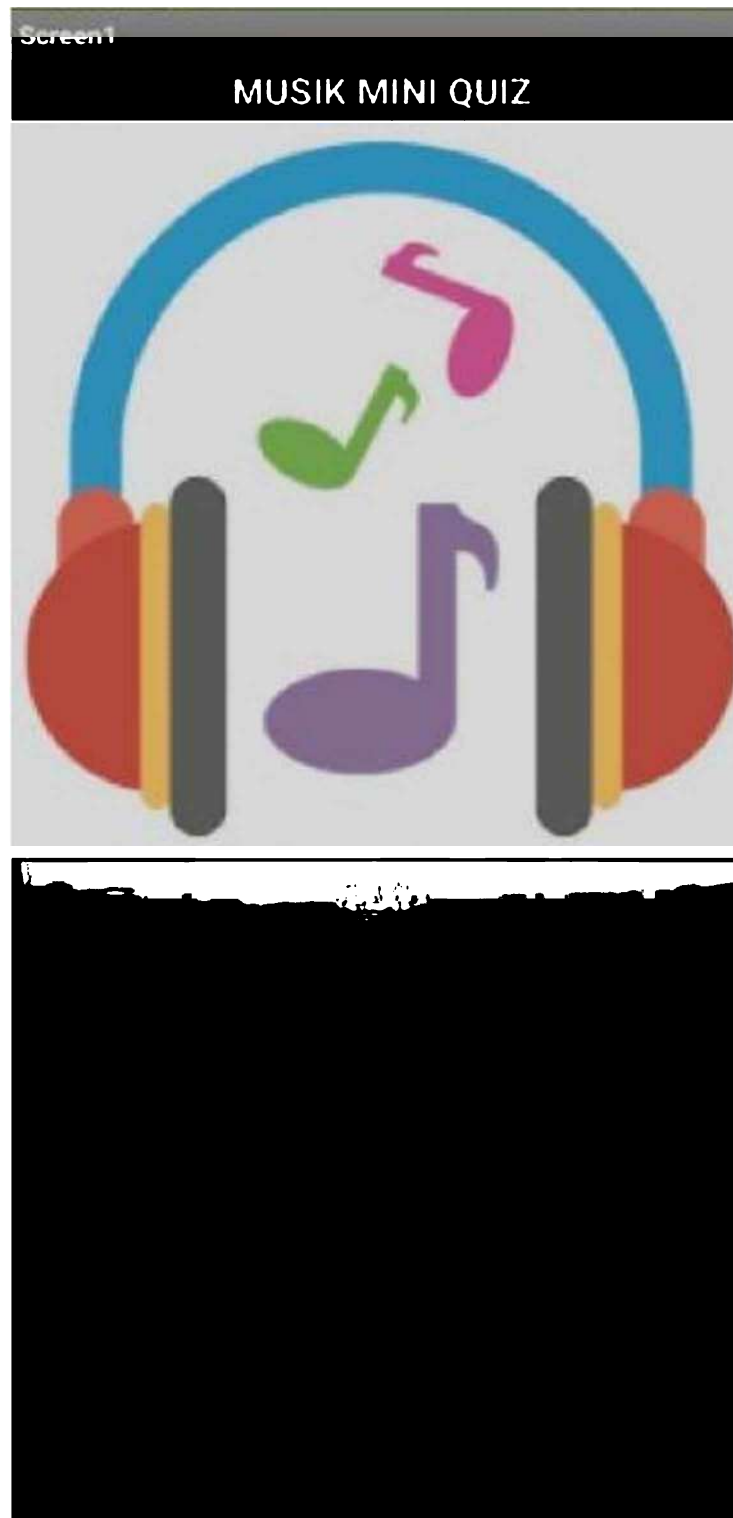
Diatas ini merupakan tampilan *dashboard* dari aplikasi yang akan aku buat. Disini aku menggunakan *tool Vertical Arrangement, Label, Image, dan button*. Kemudian aku *import* ikon untuk logo kemudian memasukkannya ke dalam *image* supaya bisa muncul ke dalam tampilan. Aku juga menambahkan *background wallpaper* supaya tampilan lebih menarik.



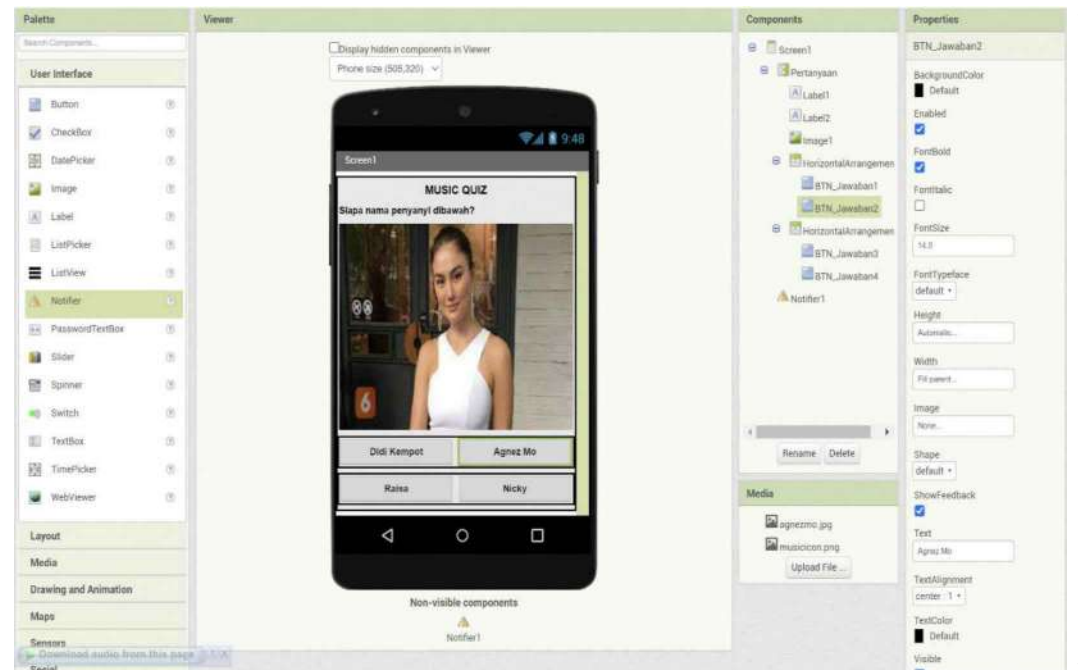
Gambar 8.9. Penjelasan *tool blocks*

Kemudian disini masuk ke menu *blocks*, menggunakan *blocks* dari *Button1 (When Button 1 Click)* kemudian menambahkan *block control* yang berfungsi untuk *open another screen*.

Berikut adalah *screenshot* dari tampilan *dashboard* yang dimunculkan lewat AI Companion simulasi *prototype* aplikasi MIT di *smartphone* Android



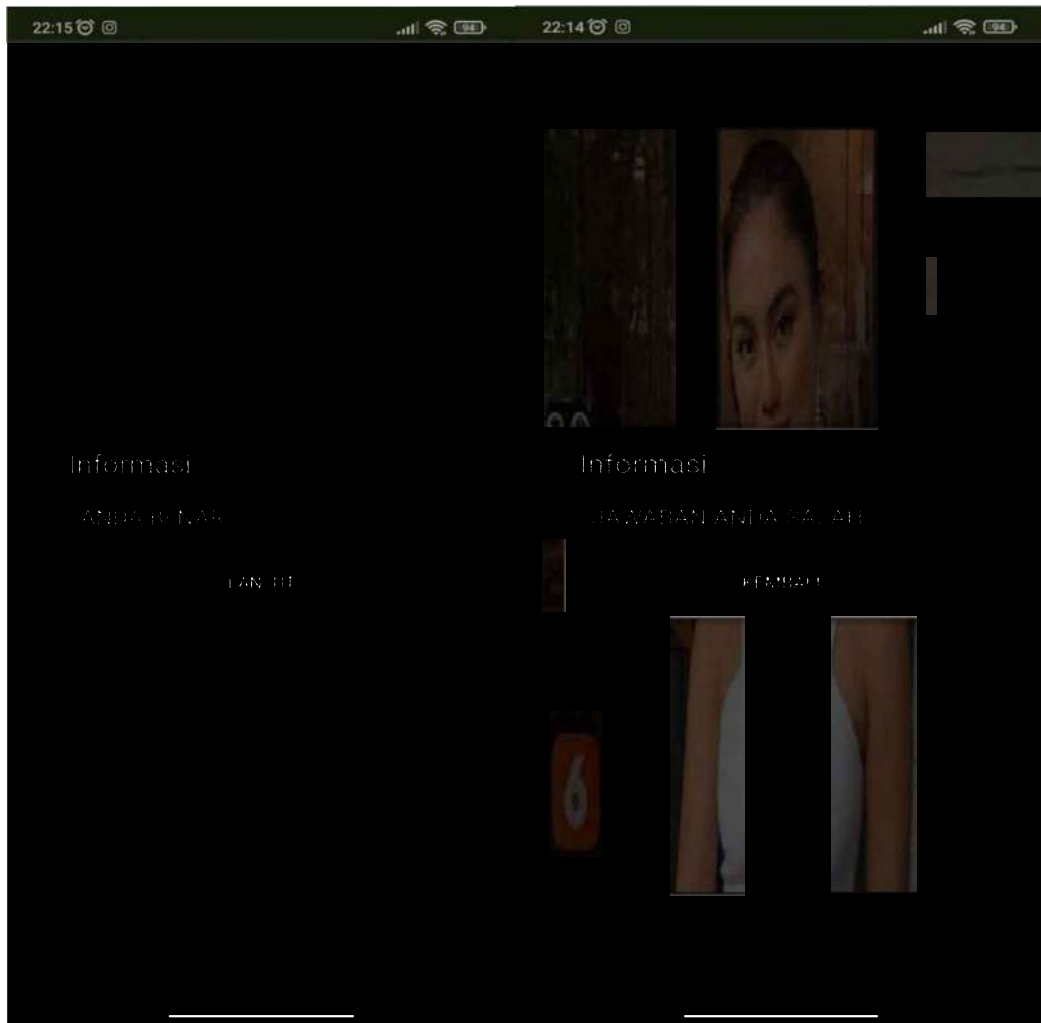
Gambar 8.10. *Screenshot* tampilan *dashboard* aplikasi



Gambar 8.11. *Design screen soal pertama*

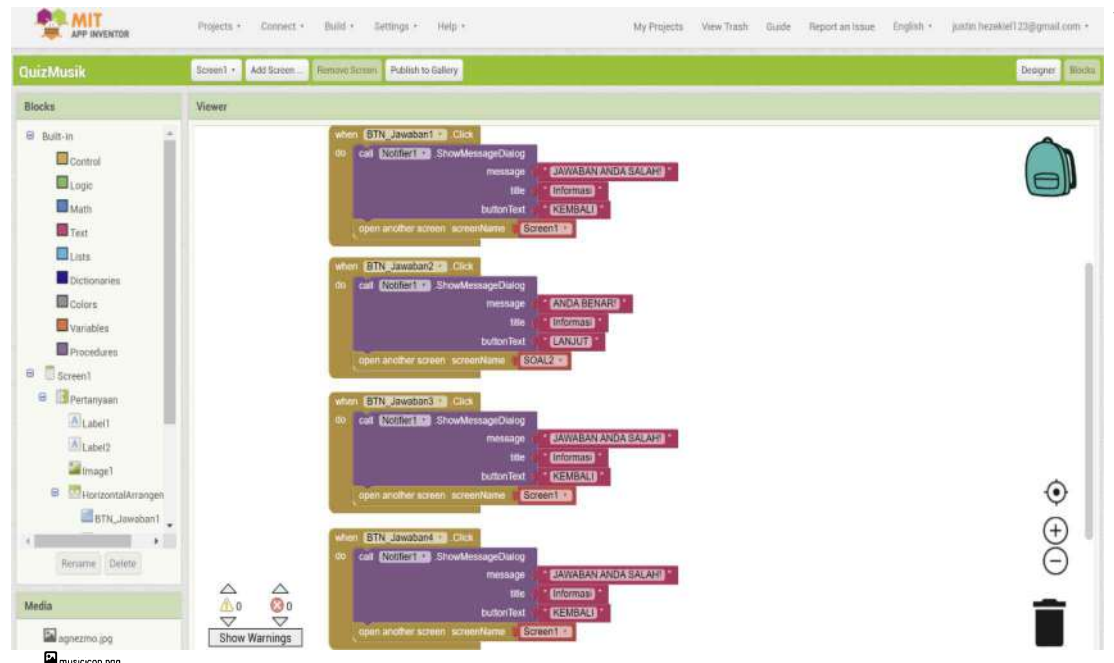
Diatas merupakan *screen* soal pertama. Aku menggunakan *layout Vertical Arrangement*, *Label*, *Image*, *Button*, *Notifier*. Pertama menggunakan *label* untuk Judul kemudian *label* kedua aku gunakan untuk soal *mini quiz*-ku. Kemudian aku menambahkan 2 *layout horizontal arrangement* kemudian memasukan *Button* untuk pilihan jawaban.

Aku menggunakan 4 *button* yang berguna untuk 4 jawaban dan hanya 1 dari 4 jawaban yang benar. Aku me-*rename Button* menjadi *BTN_Jawaban1* dan seterusnya untuk memudahkan saat mengatur *block* yang aku butuhkan Kemudian jika jawaban salah maka akan muncul *notifier* “JAWABAN ANDA SALAH!” kemudian Kembali ke *screen* soal jika jawaban benar muncul *notifier* “ANDA BENAR” kemudian lanjut ke pertanyaan berikutnya.



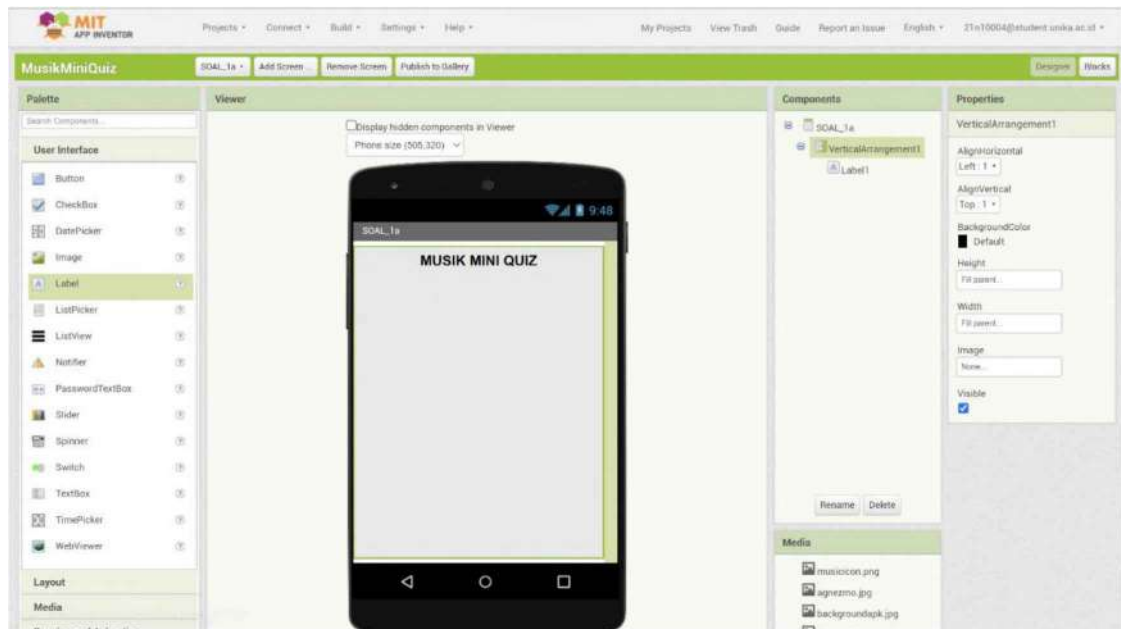
Gambar 8.12. *screenshot* dari *screen* soal pertama

Diatas merupakan *screenshot* tampilan dari aplikasi saat muncul *notifier*. Yang kiri merupakan notifikasi jika jawaban salah, yang kanan merupakan notifikasi jika jawaban benar.



Gambar 8.13. Penjelasan *block* di soal 1

Berikut merupakan *block* yang aku gunakan untuk *screen* soal 1. Menggunakan *block* when BTN_Jawaban click call *Notifier ShowMessageDialog* untuk memunculkan notifikasi jika salah akan muncul JAWABAN ANDA SALAH, jika benar akan muncul notifikasi ANDA BENAR. Kemudian menggunakan *block* open another screen jika salah akan kembali ke screen soal sebelumnya jika benar maka akan lanjut ke screen berikutnya.

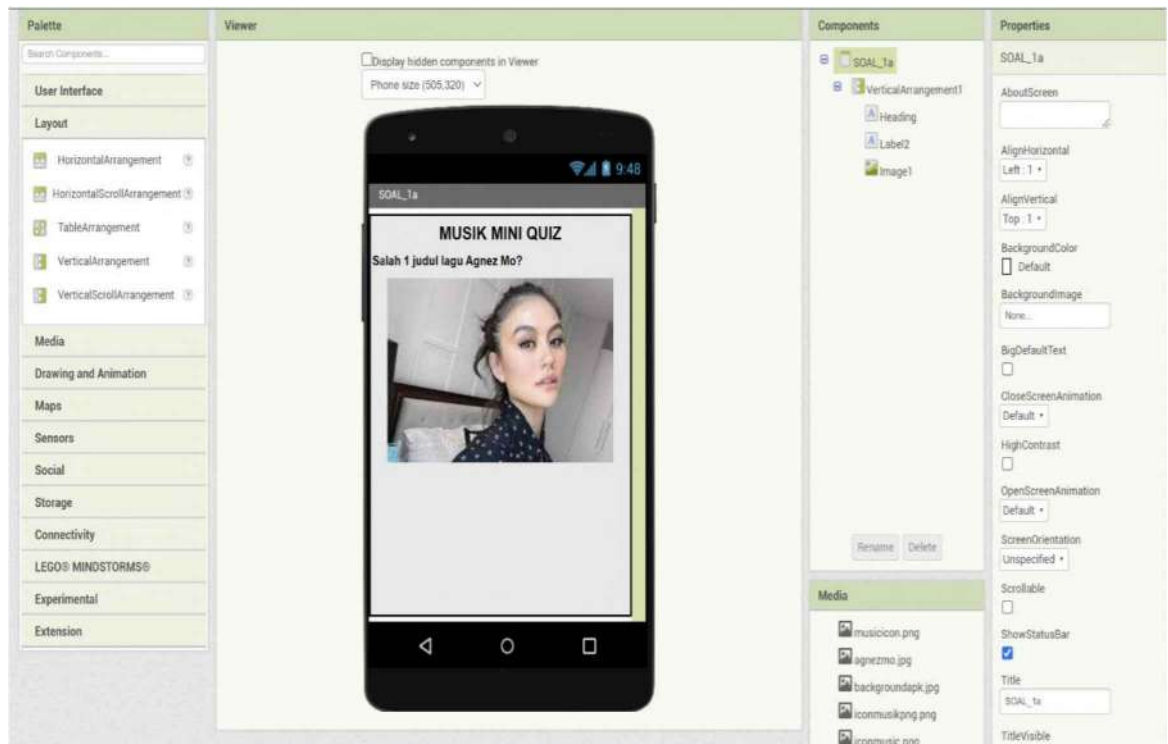


Gambar 8.14. Penjelasan *design screen* soal 1a

Masuk ke *screen* soal 1a disini masih menggunakan tema yang sama yaitu penyanyi Agnez Mo. *Tool* yang akan aku gunakan ada *layout vertical*, *horizontal arrangement*, *image*, *button*, *label*, dan *notifier*.

Untuk *design screen*nya tidak berbeda dengan *screen* sebelumnya aku memasukan *layout vertical arrangement* kemudian untuk *height* dan *width*nya dirubah ke *fill parent*.

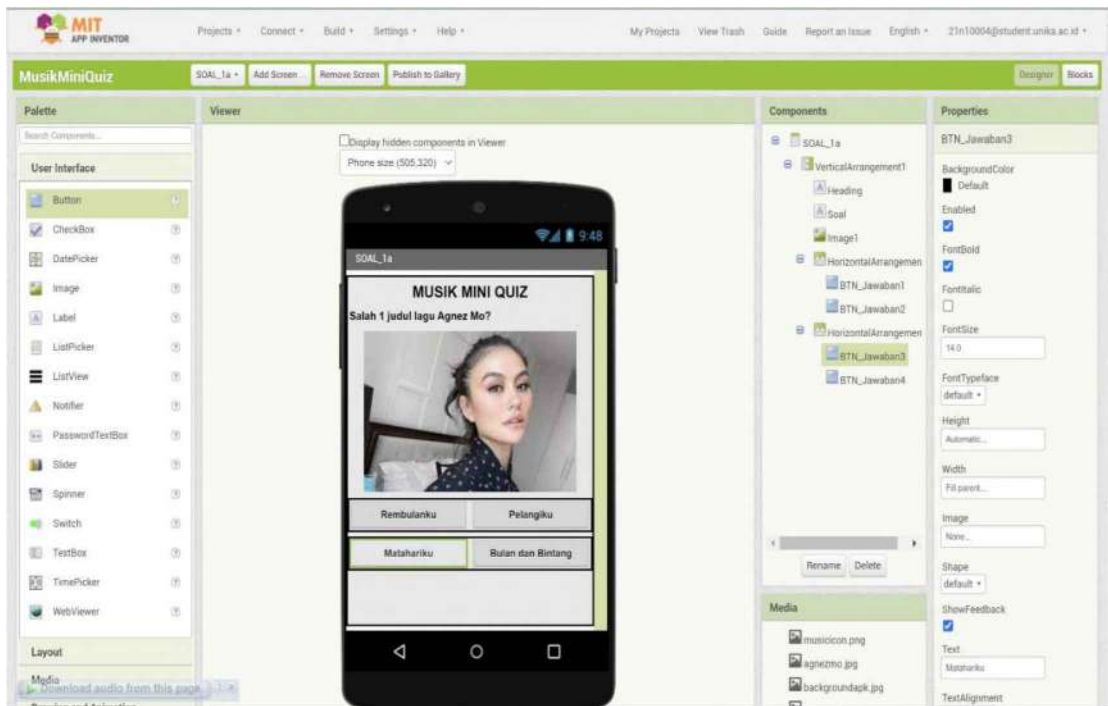
Kemudian aku memasukan *label* untuk judul kemudian *text*nya aku ganti menjadi “MUSIK MINI QUIZ” dengan *font bold* dan *font size* 20 *width* aku merubah ke *fill parent* dan *text alignment*-nya aku rubah ke *center*.



Gambar 8.15. *Design screen soal 1a*

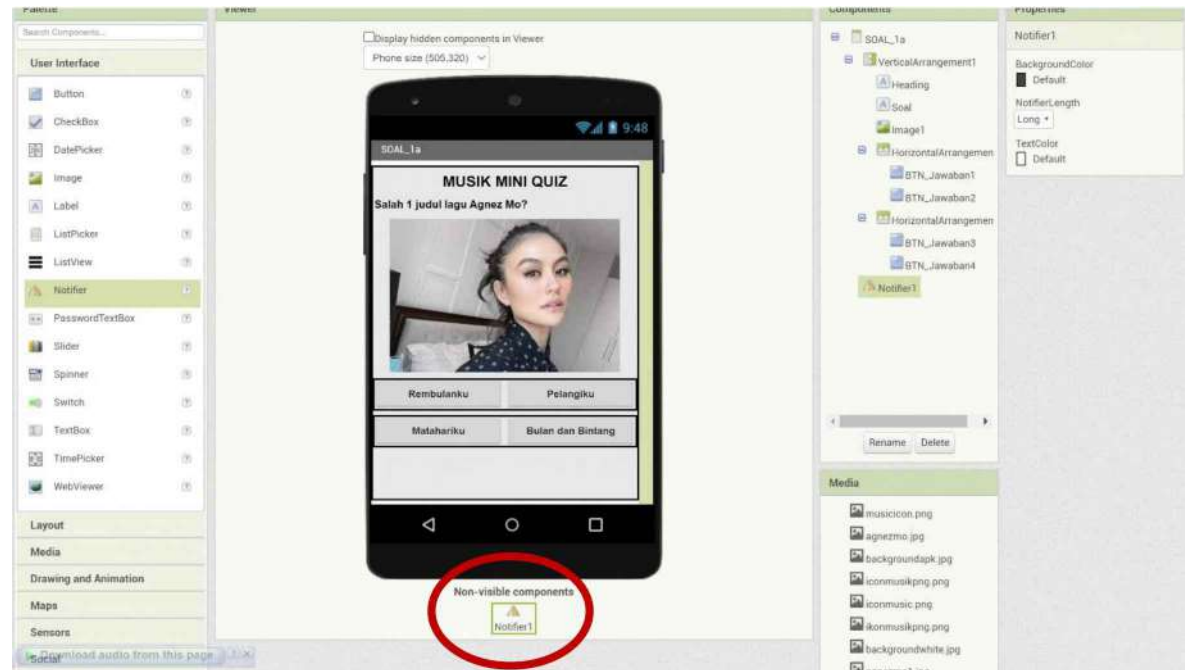
Selanjutnya untuk *screen* Soal 1a aku memasukan *label* lagi untuk menulis soal. Aku memasukan soal seputar judul lagu Agnez Mo dengan *font bold* dan *font size 15*.

Kemudian aku menambahkan *image* kemudian *import image* dari *device* yang digunakan kemudian memasukkannya kedalam *tool image* lewat *picture menu*. Kemudian aku mengatur *width-nya* dengan *fill parent*.



Gambar 8.16. *Design tool screen 1a*

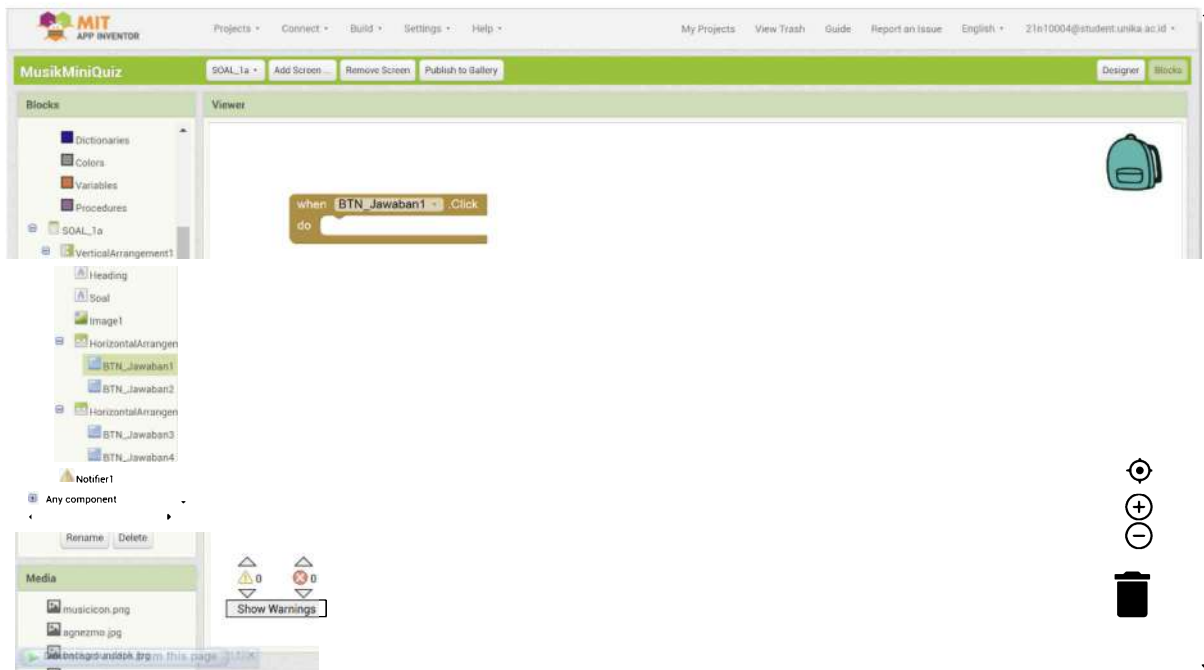
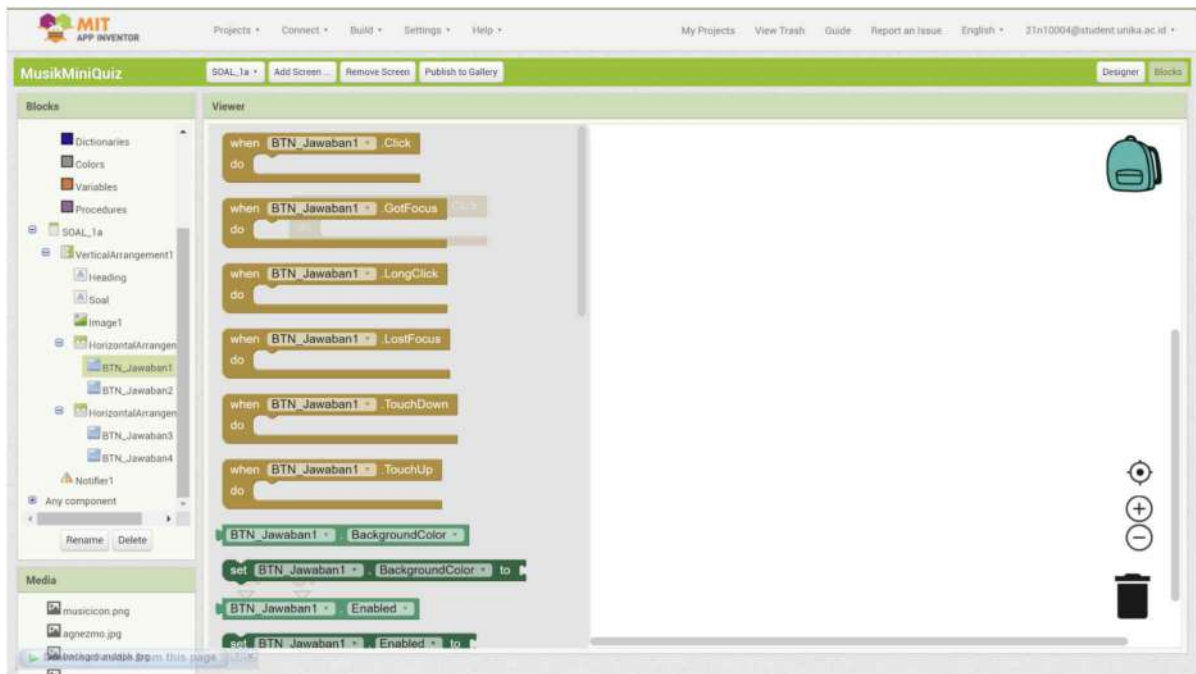
Kemudian aku menambahkan *horizontal arrangement* kemudian aku tambahkan *tool button* sebanyak 4 buah untuk pilihan jawaban kemudian untuk *width button* aku atur *fill parent*. Aku me-*rename tool button* menjadi BTN_Jawaban1, BTN_Jawaban2, BTN_Jawaban3, BTN_Jawaban4 untuk memudahkan saat mengatur *block*-nya. Kemudian aku ganti *text* di *button* kemudian aku *text bold* juga. Aku menyiapkan 4 jawaban dan hanya ada 1 diantaranya yang benar.



Gambar 8.17. Penggunaan *notifier* pada *screen 1a*

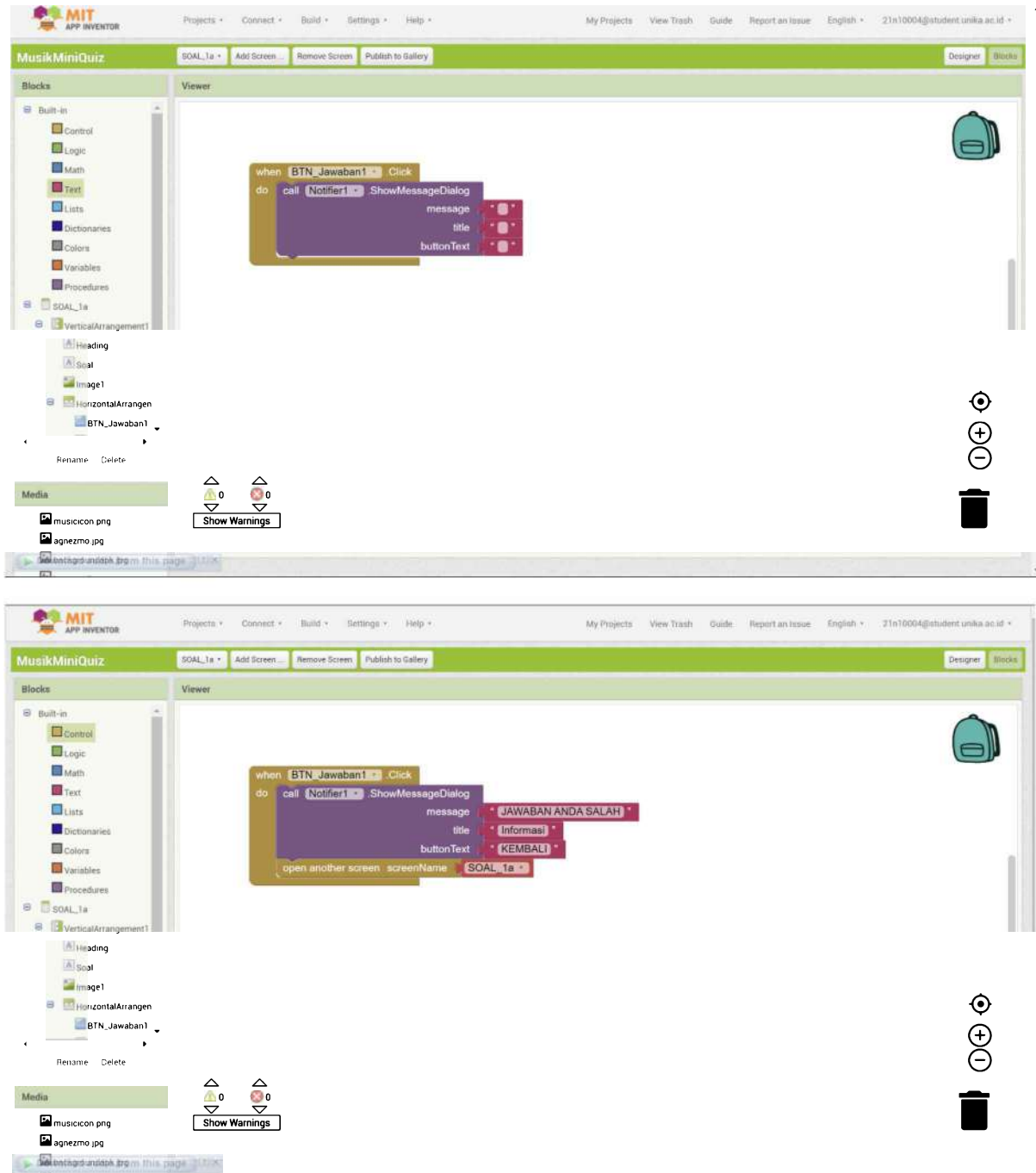
Kemudian terakhir untuk design aku menambahkan *notifier tool* juga yang termasuk *non-visible component*, atau komponen yang tidak terlihat secara langsung dalam *design interface*.

Kemudian untuk bagian *block* disini aku menggunakan *block* dari *button* BTN_Jawaban1.



Gambar 8.18. Penjelasan *block* untuk *screen 1a*

Disini karena pada jawaban1 aku ingin jawabannya salah maka aku akan memanggil *notifier* kemudian menggunakan *block* yang *show message dialog* untuk memunculkan notifikasi jika *user* memilih jawaban yang salah.



Gambar 8.19. Penjelasan *block* untuk *screen* soal 1a

Kemudian dibagian *block notifier* dibagian *message title* dan *buttonText* aku menambahkan *empty text* untuk memunculkan pesan yang ingin aku munculkan saat jawabannya salah. Kemudian aku menggunakan *control block* untuk menambahkan *block open another screen*. Karena jika

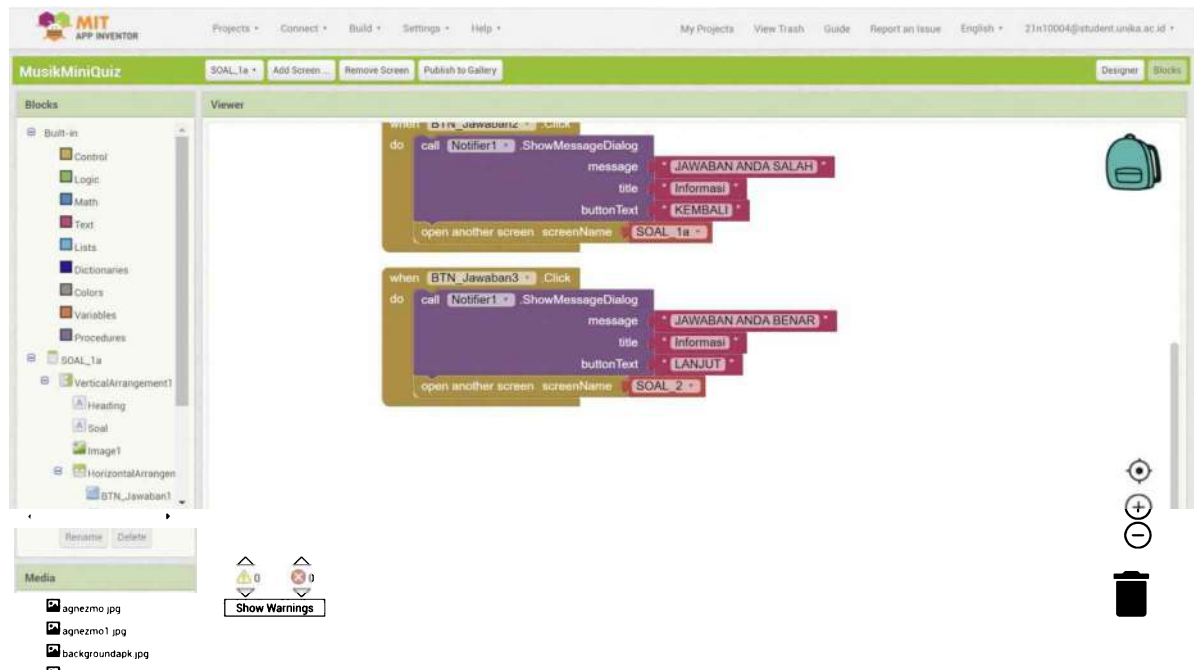
salah, *user* akan kembali ke soal tersebut maka di *screen* SOAL_1a ini jika *user* salah akan kembali ke *screen* SOAL_1a.



Gambar 8.20. Penjelasan *block* untuk *screen* soal1a

Untuk BTN_Jawaban2 karena jawaban ini salah maka *block*-nya tidak berbeda dengan *block* untuk BTN_Jawaban1.

Kemudian untuk BTN_Jawaban3 karena dipilihan ini aku ingin jawaban benar maka pada bagian *message* dan *button* text dapat diubah di-*block* ini aku mengubah *message* menjadi (JAWABAN ANDA BENAR) kemudian *buttonText*nya diubah menjadi (LANJUT) dan bagian *open another screennya* diubah ke SOAL_2.

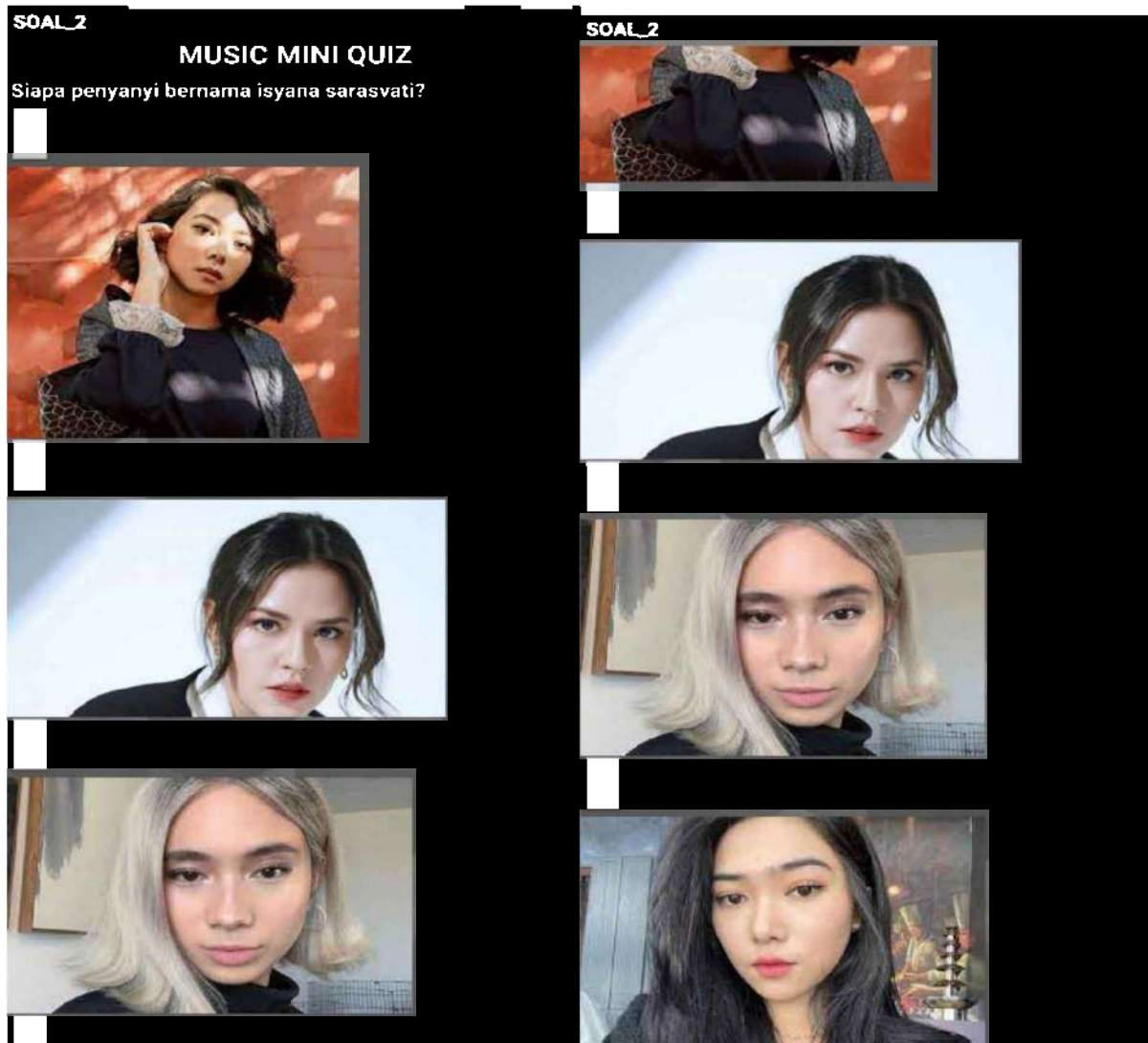


Gambar 8.21. Penjelasan *block* untuk *screen 1a*



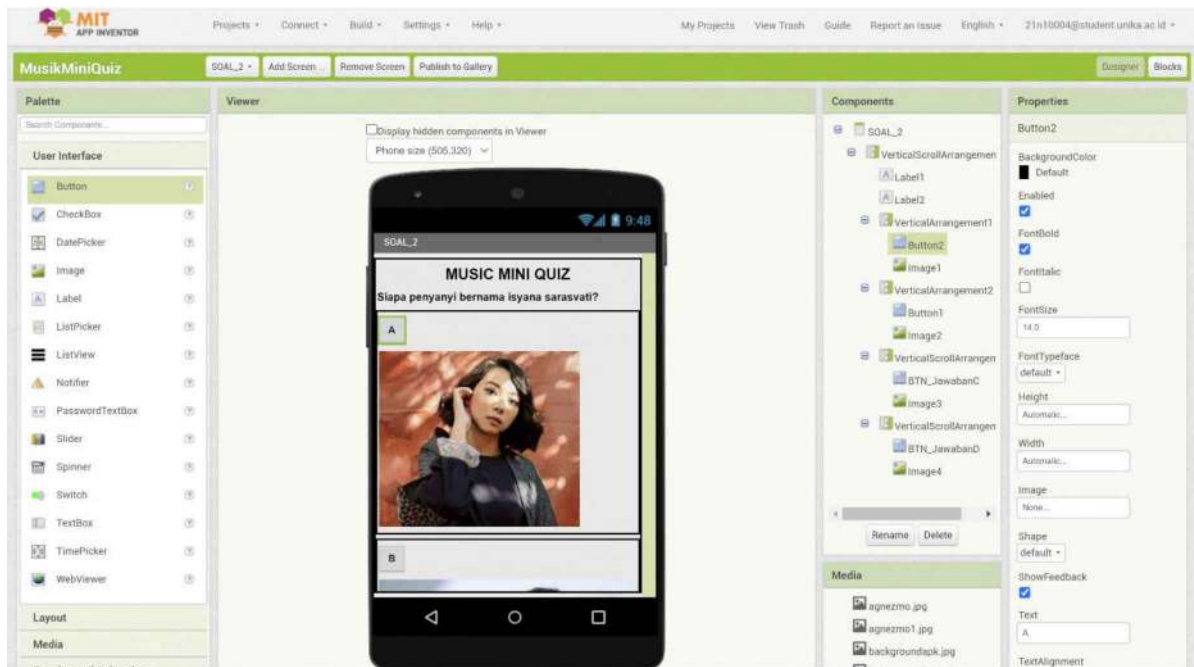
Gambar 8.22. Penjelasan *screen* untuk soal 1a

Untuk BTN_Jawaban4 disini karena jawabannya juga salah jadi *block*-nya sama dengan *block* untuk BTN_Jawaban1 dan BTN_jawaban2.



Gambar 8.23. *Screenshot tampilan screen soal 2*

Kemudian untuk soal 2 aku menggunakan *quiz* dengan tema pilihan bergambar untuk tampilan di Android-nya seperti dibawah ini.



Gambar 8.24. Design screen soal2

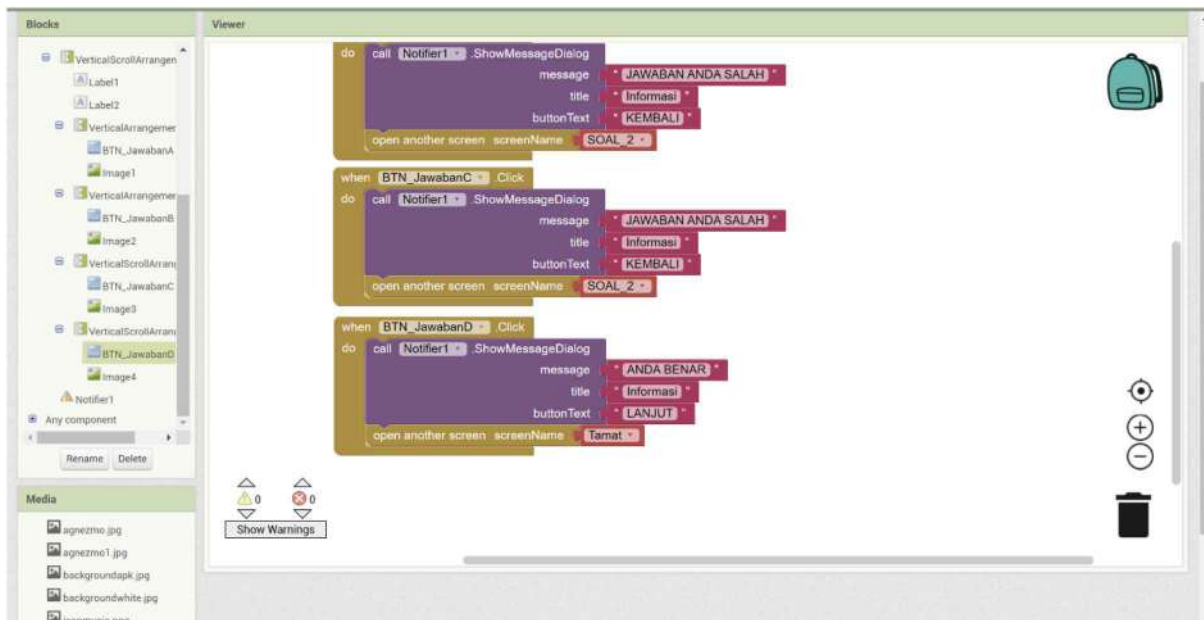
Untuk *design* detailnya disini untuk *screen* aku menggunakan *VerticalScrollArangement* supaya *screen* bisa di-*scroll* kebawah karena akan menggunakan banyak foto, kemudian aku menggunakan *label* dengan *font* 20 dan *bold* untuk *headingnya*. Kemudian aku menambahkan label lagi dengan *font size* 15 di-*bold* untuk pertanyaannya. Disini pertanyaannya adalah menebak wajah artis bernama Isyana Sarasvati.

Kemudian aku menggunakan *VerticalArrangement* kemudian ditambah *image* dan *button* untuk pilihan jawabannya. Kemudian aku juga me-*rename* *button* menjadi *BTN_JawabanA*, *BTN_JawabanB*, *BTN_JawabanC*, *BTN_JawabanD* supaya memudahkan saat mengatur *block*. Terakhir aku menambahkan *notifier* supaya dapat memunculkan notifikasi saat *user* menjawab salah atau benar.



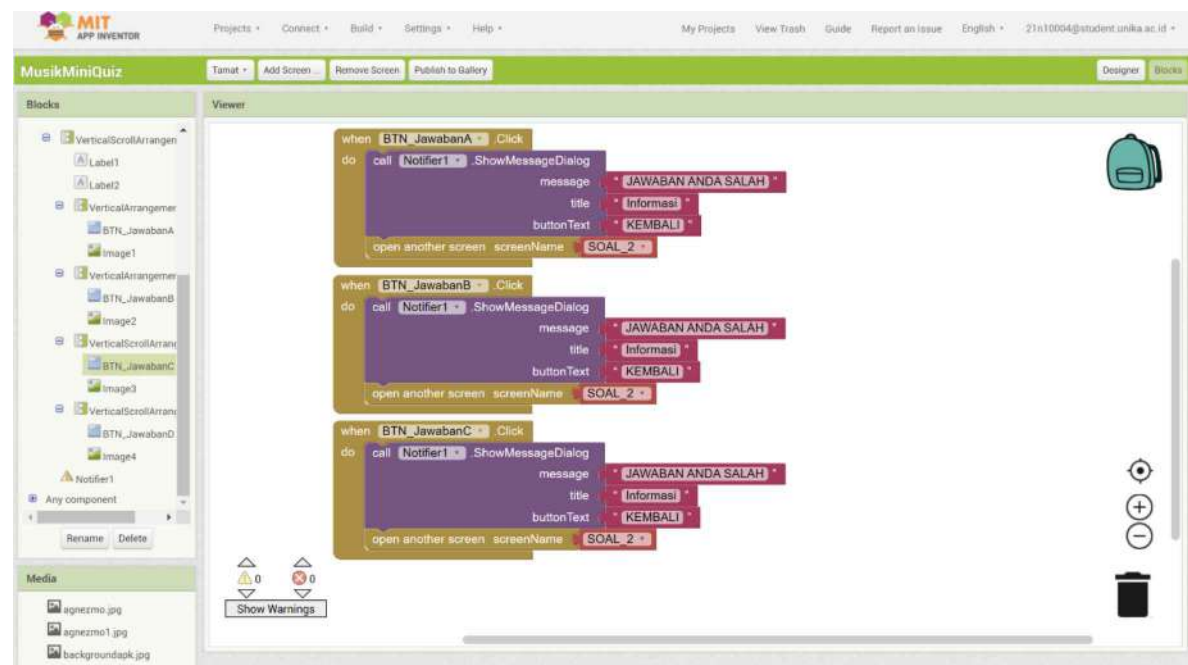
Gambar 8.25. Penjelasan *block* untuk *screen* soal2

Disini untuk *block* pertama BTN_JawabanA karena jawabannya salah tidak berbeda dengan *screen* sebelumnya aku menggunakan *block button when* BTN_JawabanA *click* kemudian menambahkan *block notifier show message dialog* juga memasukan *empty text* untuk mengatur memunculkan “JAWABAN ANDA SALAH” juga *button* “KEMBALI” untuk Kembali ke soal sebelumnya.



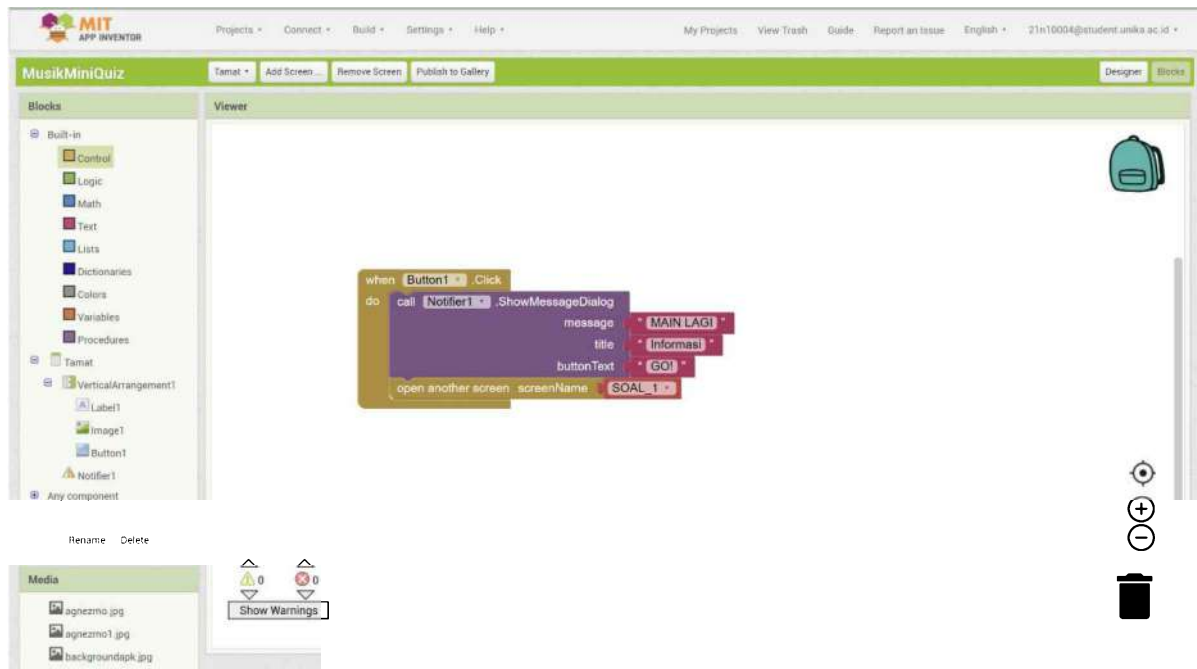
Gambar 8.26 Penjelasan *block* untuk screen soal2

Kemudian karena jawaban B dan C juga salah maka BTN_JawabanB dan BTN_JawabanC menggunakan *blocks* yang sama juga seperti BTN_JawabanA.



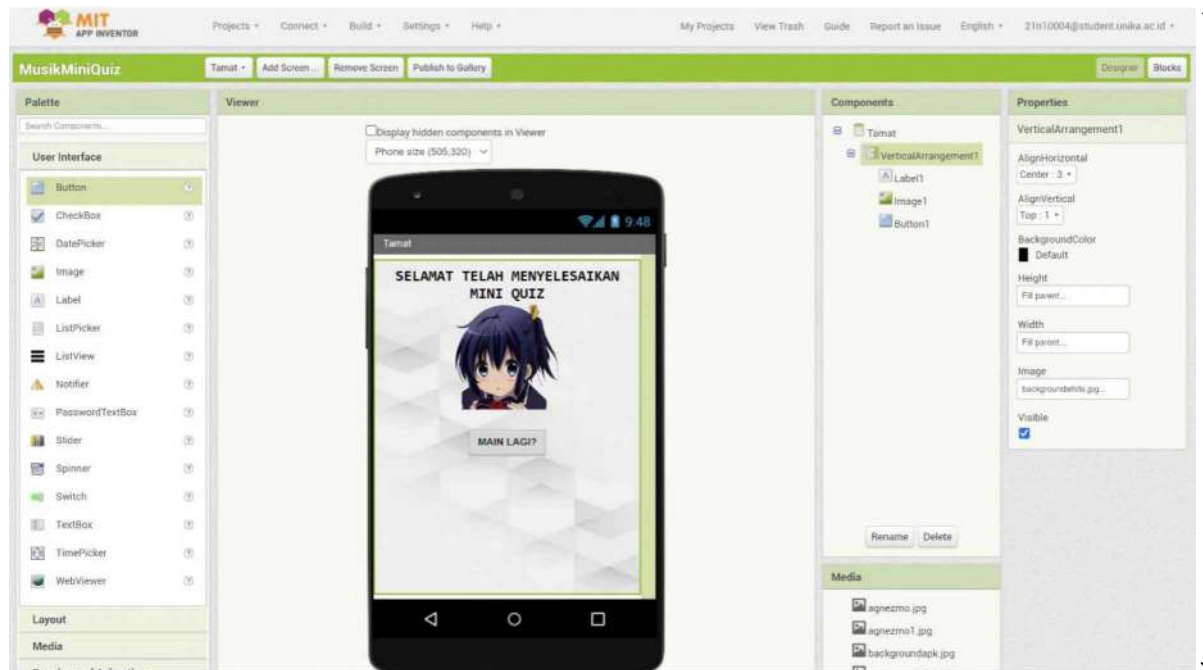
Gambar 8.27. Penjelasan *blocks* untuk screen soal2

Disini untuk `BTN_JawabanD` karena benar maka *message*-nya “ANDA BENAR” dan *button text*nya lanjut ke *screen* selanjutnya.



Gambar 8.28. Penjelasan *block* untuk *screen* penutup

Untuk *screen* terakhir desainnya aku menggunakan *label* untuk *heading* dengan *font size* 20 dengan *bold* bertuliskan “SELAMAT TELAH MENYELESAIKAN MINI QUIZ”. Kemudian bawahnya menggunakan *button* bertuliskan “MAIN LAGI?”. Juga menggunakan *tool notifier*.



Gambar 8.29. *Design* untuk *screen* penutup

Diatas merupakan *blocks* yang aku gunakan untuk *screen* terakhir disitu aku menggunakan *blocks when Button1 click* maka akan muncul *notifier* “MAIN LAGI” dan *button text* “GO” dan kembali ke *screen* soal 1. Berikut adalah *tutorial* singkat atau *walkthrough*-ku saat membuat *mini game quiz* dengan tema musik menggunakan MIT App Inventor.

Hasil jadi

https://drive.google.com/file/d/1QUHfyt8-4kQc_GG5HwMapRpTmiTKBGud/view?usp=sharing

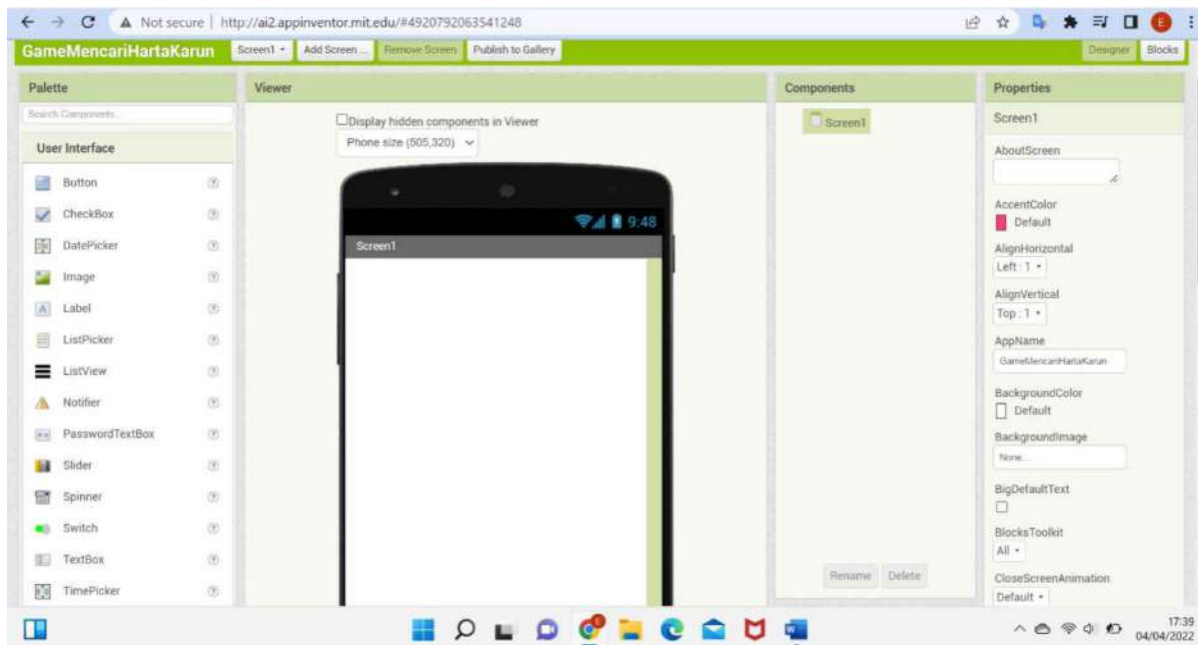
Membuat Game Menemukan Harta Karun

Enrico Natanael N

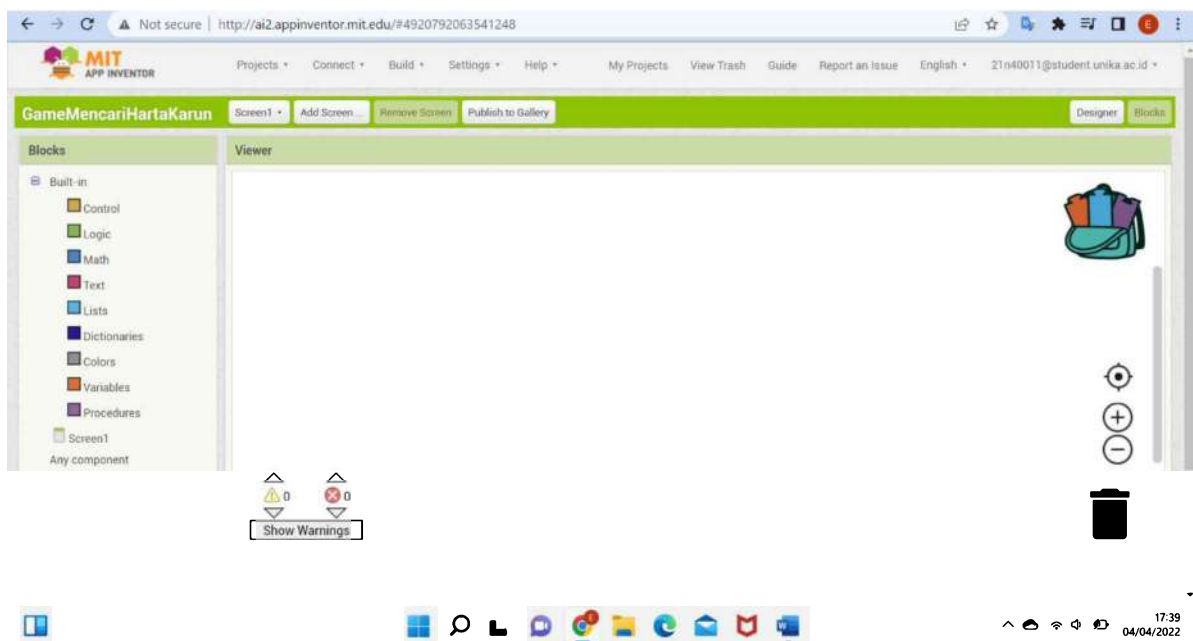
Hari ini kita akan belajar membuat *game* sederhana dengan menggunakan aplikasi MIT App Inventor. Dengan aplikasi MIT App Inventor ini kita dapat dengan mudah membuat *game* tanpa harus menuliskan *coding*. Selain sederhana dan mudah digunakan, aplikasi ini juga mempunyai tampilan desain yang menarik. Semua fasilitas itu bisa kita gunakan dengan menggunakan klik *drag and drop* menggunakan *block* perintah. Membuat *game* dengan MIT App Inventor ini tidaklah sulit, kita hanya memerlukan ide dan kreatifitas yang tinggi agar dapat membuat sebuah permainan yang menyenangkan bagi si pengguna. Nah kali ini aku akan bagikan *tutorial* dan tips untuk membuat *game*.

Pengenalan MIT

Sebelum masuk ke cara membuatnya, aku mau jelasin ke kalian dulu nih fungsi dari komponen-komponen yang ada di MIT App Inventor. Mari kita mulai dari halaman awalnya terlebih dahulu. MIT App Inventor memiliki dua halaman utama yaitu halaman *designer* sebagai halaman tampilan aplikasi dan yang kedua yaitu halaman *blocks* yang berfungsi untuk menjalankan programnya. Berikut ini adalah tampilannya:



Gambar 9.1. Halaman *designer*



Gambar 9.2. Halaman *blocks*

Penjelasan:

1. Halaman Designer

Di halaman designer ini terdapat beberapa *tools* seperti *Palette*, *Viewer*, *Components*, *Media*, dan *Properties*. *Tools*

tersebut mempunyai fungsi untuk mendesain tampilan aplikasi android yang akan kita buat sesuai keinginan. Berikut ini adalah penjelasannya:

- a. *Palette* merupakan jendela atau tempat untuk mengambil komponen-komponen seperti *User Interface, Layout, Media, Drawing and Animation, Maps, Sensors, Social, Storage, Connectivity, LEGO MINDSTORMS, Experimental*, dan *Extension*.
- b. *Viewer* merupakan tempat untuk melihat dan mengatur tampilan komponen pada aplikasi kita nantinya.
- c. *Components* merupakan tempat yang berfungsi untuk mengatur komponen-komponen yang telah kita letakkan di *viewer*.
- d. *Properties* merupakan tempat untuk mengatur properti dan komponen yang kita gunakan, seperti tinggi, lebar, warna, besar huruf, dan lain-lain.
- e. *Media* merupakan tempat untuk mengunggah gambar untuk kita gunakan pada aplikasi kita nantinya.

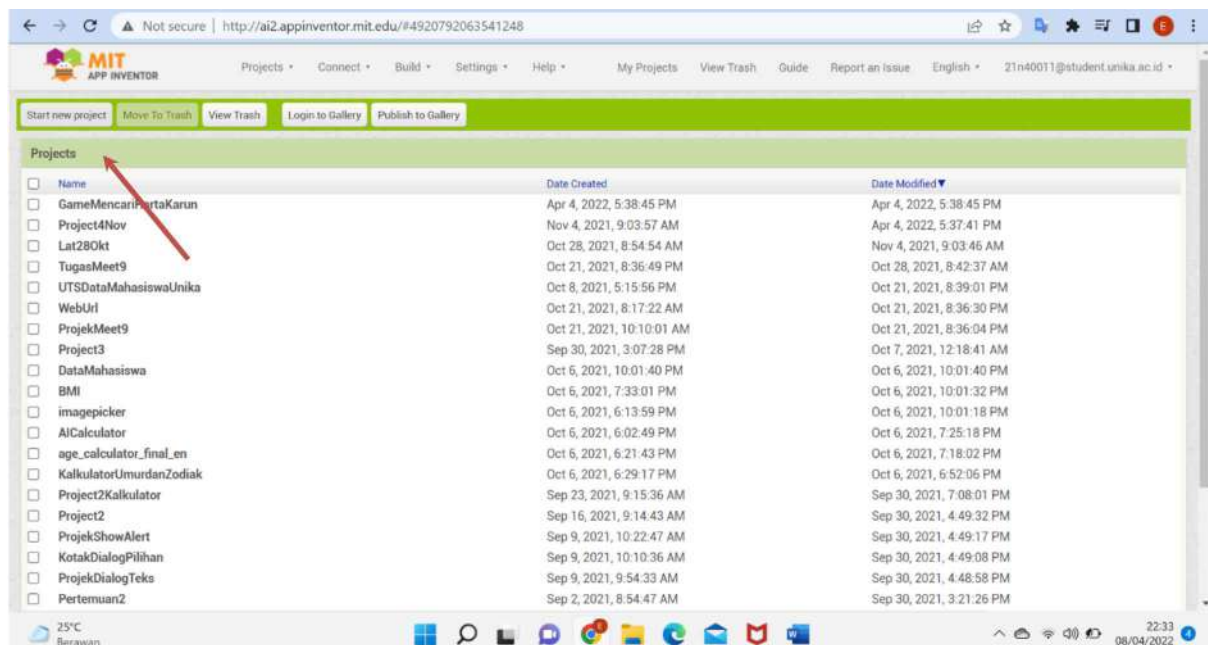
2. Halaman *Blocks*

Di halaman *blocks* ini terdapat berbagai blok kode untuk menjalankan program dengan fungsinya masing-masing. Komponen-komponen yang terdapat pada halaman *blocks* ini yaitu *Control, Logic, Math, Text, Lists, Colors, Variables*, dan *Procedures* yang memiliki fungsi sebagai berikut:

- a. *Control* merupakan blok kode yang mempunyai fungsi untuk membuat kontrol, misalnya jika "x" maka akan ke step "y".
- b. *Logic* merupakan blok kode yang mempunyai fungsi untuk membuat logika *true/false*.
- c. *Math* merupakan blok kode yang mempunyai fungsi perhitungan matematika, seperti penambahan, pengurangan, perkalian, pembagian, dan lainnya.
- d. *Text* merupakan blok kode yang mempunyai fungsi untuk menampilkan teks yang nanti akan muncul di halaman desain kita.
- e. *Lists* merupakan blok kode yang mempunyai fungsi di lingkup *list*.
- f. *Colors* merupakan blok kode yang mempunyai fungsi untuk mengatur warna.
- g. *Variables* merupakan blok kode yang mempunyai fungsi untuk mengatur *variable-variabel* dalam pemrograman.
- h. *Procedures* merupakan blok kode yang mempunyai fungsi untuk membuat suatu prosedur dan suatu fungsi yang mengembalikan ke hasil tertentu.

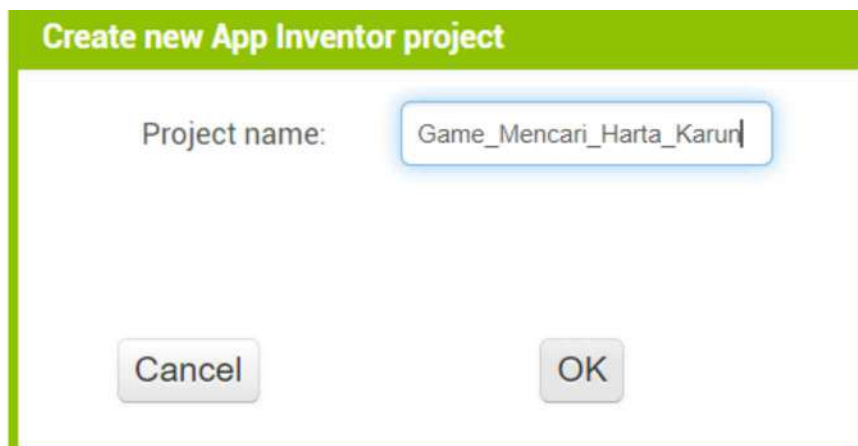
Membuat Game

1. Pilih *Start New Project*.



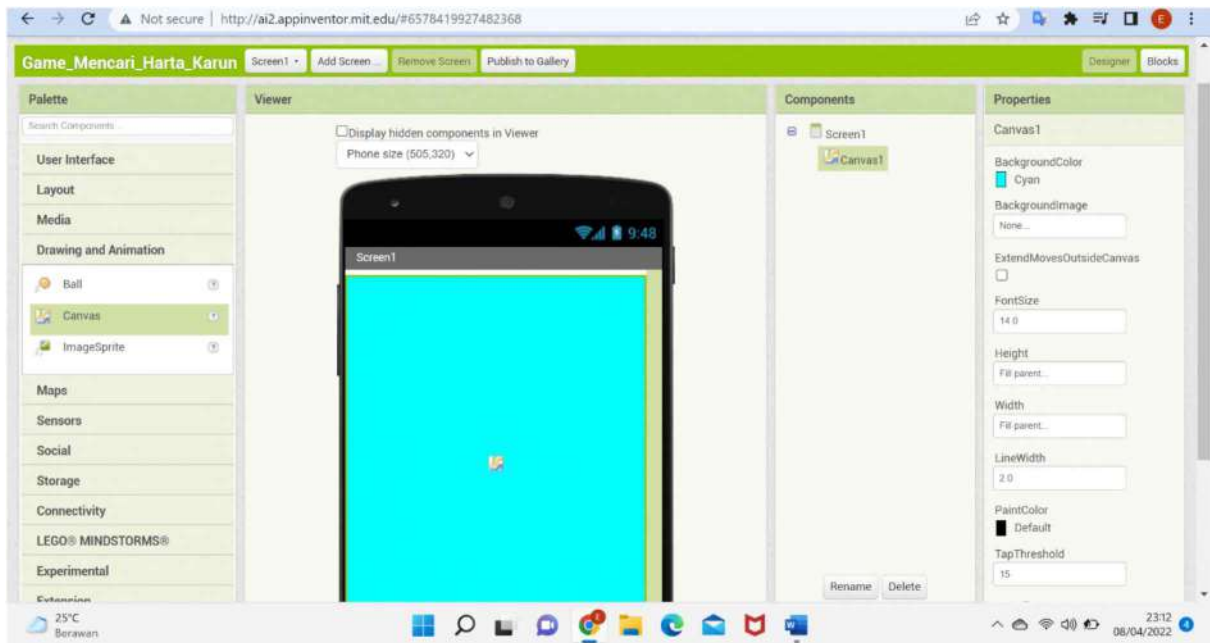
Gambar 9.3. halaman awal

2. Isikan nama project *game* kita.



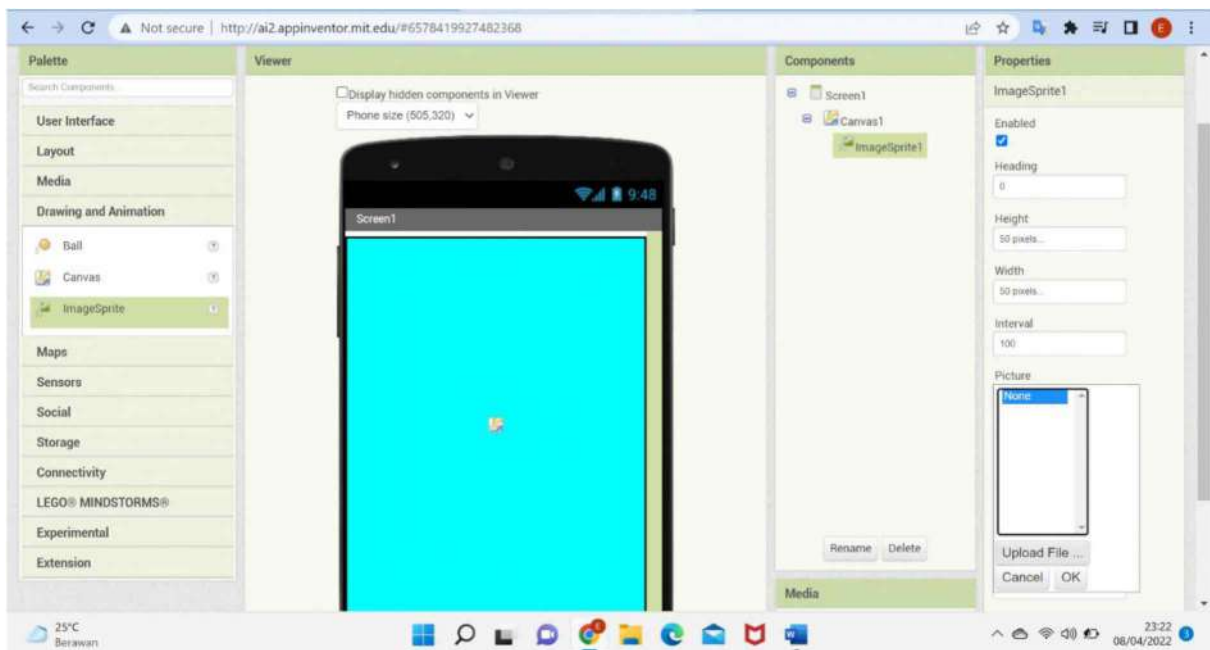
Gambar 9.4. halaman *Project Name*

3. Masukkan *canvas* dengan cara menyeret dan lepaskan di area desain, lalu atur *background color*, *height*, dan *width* seperti di gambar.



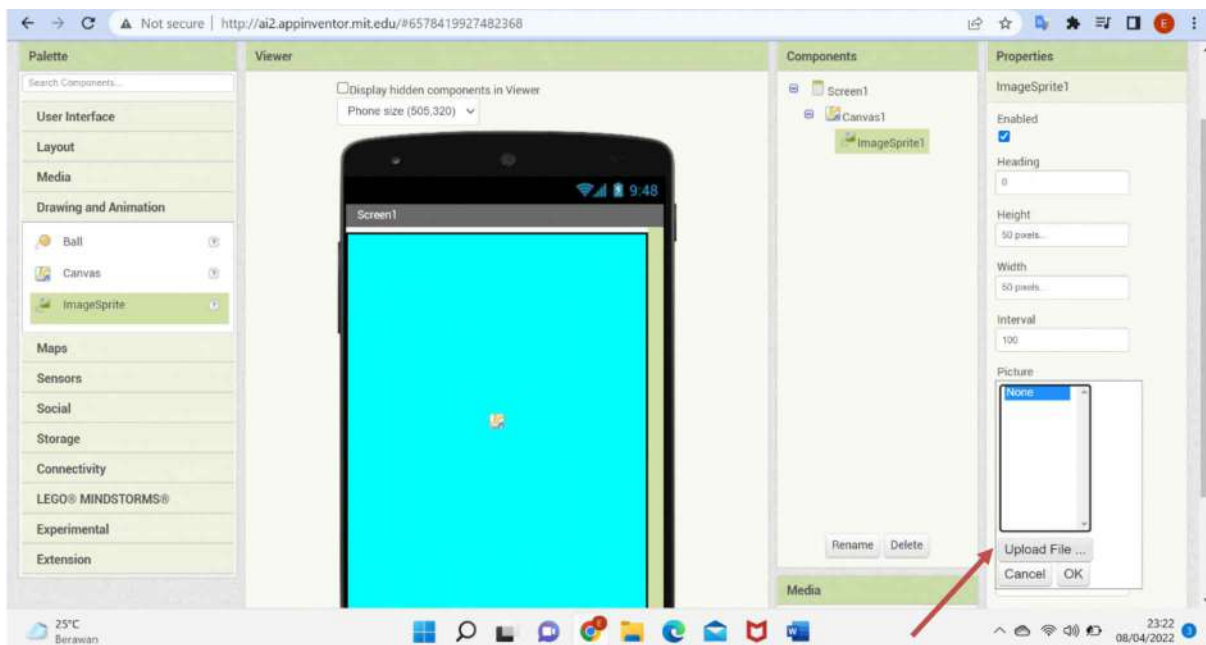
Gambar 9.5. Tampilan halaman *designer*

4. Masukkan *ImageSprite* lalu atur *height* dan *width* seperti di gambar.



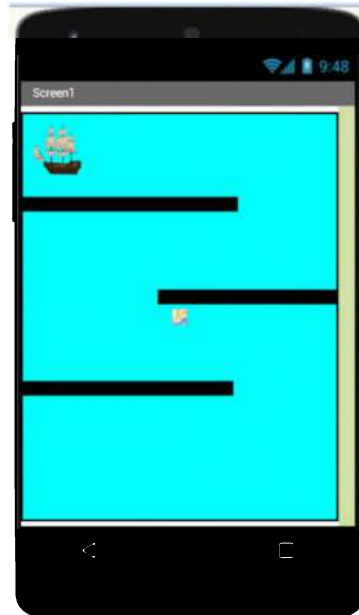
Gambar 9.6. Tampilan halaman *designer*

5. Untuk memasukkan gambar kapal, silahkan *download* terlebih dahulu dan *upload* gambar di bagian *picture*.



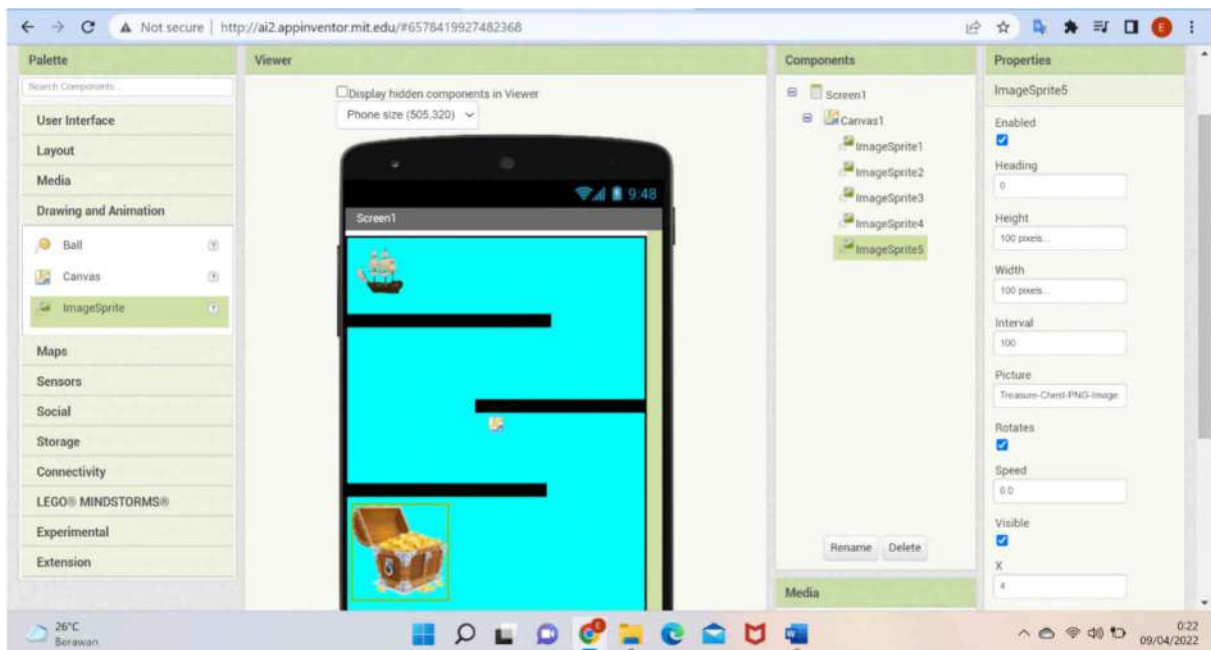
Gambar 9.7. Tampilan halaman *designer*

6. *Download* gambar garis hitam sebagai rintangan *game* dan gambar harta karun, lalu *upload* gambar seperti di *step* ke 6.
7. Masukkan *ImageSprite* dan gambar garis hitam lalu buatlah rintangan seperti gambar di bawah ini.



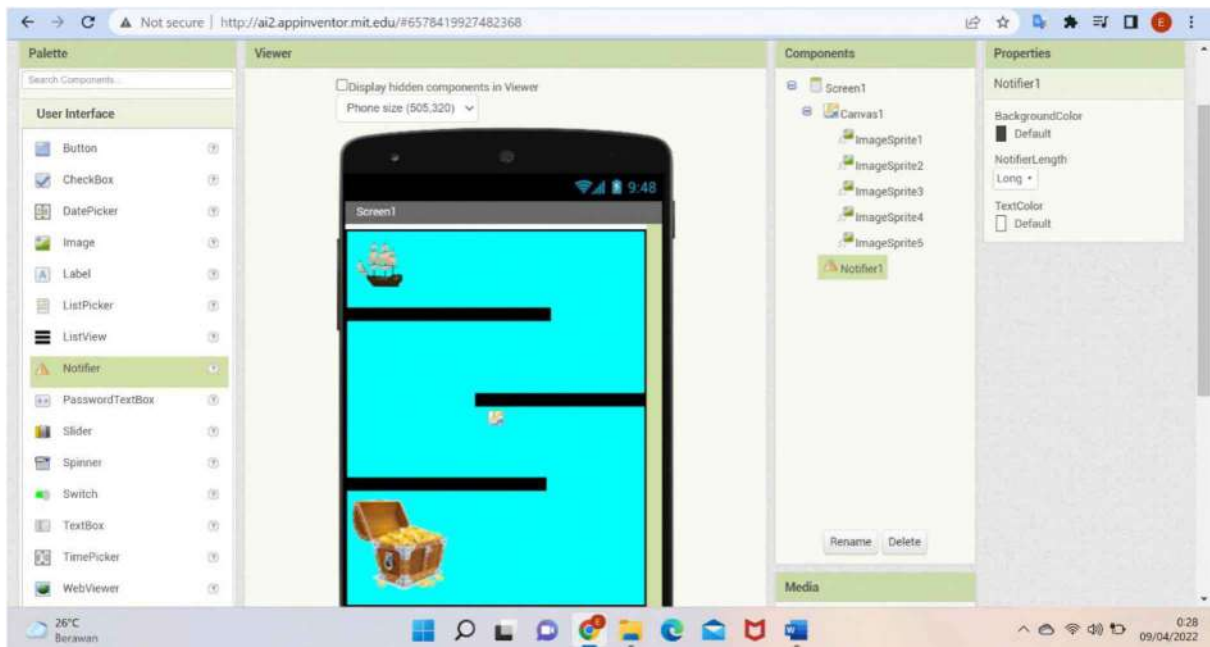
Gambar 9.8. Tampilan halaman *designer*

8. Masukkan ImageSprite dan gambar harta karun lalu atur height dan width seperti gambar di bawah ini.



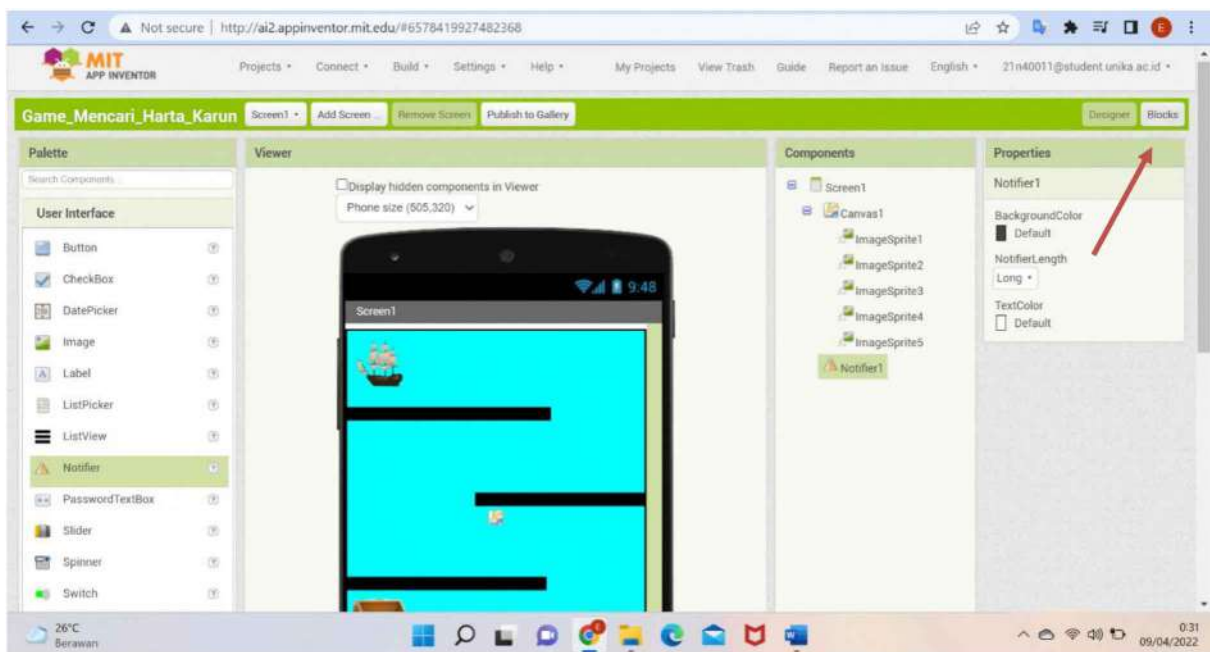
Gambar 9.9. Tampilan halaman *designer*

9. Masukkan *notifier*.



Gambar 9.10. Tampilan Halaman *designer*

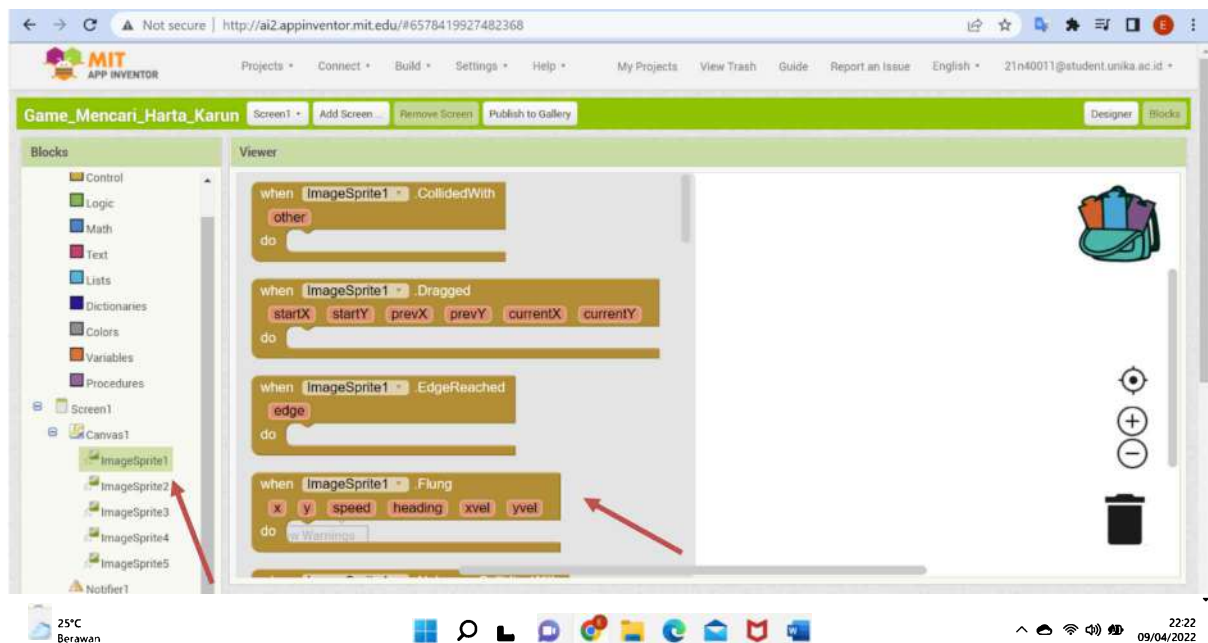
10. Kita masuk ke bagian *blocks*, silahkan klik *blocks* di bagian kanan atas.



Gambar 9.11. Tampilan untuk menuju halaman *blocks*

11. Klik *ImageSprite1* atau *ImageSprite* kapal untuk menampilkan *block-block* perintah *ImageSprite1*, pilih *block*

perintah *when ImageSprite1.Flung* seret dan lepaskan di area kerja.

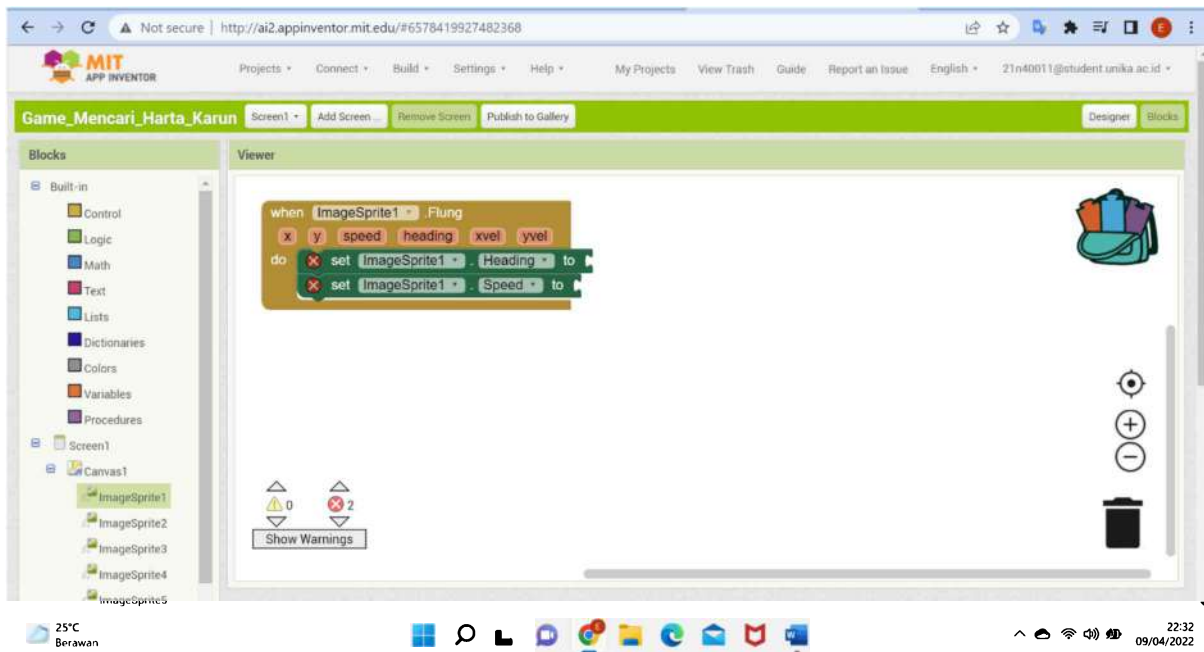


Gambar 9.12. Tampilan halaman *blocks*

12. Klik lagi *ImageSprite1*, dan cari *block* perintah *set ImageSprite1.Heading to* dan *set ImageSprite1.Speed to*.

set ImageSprite1.Heading to = *block* perintah untuk arah gerak kapal

set ImageSprite1.Speed to = *block* perintah untuk kecepatan kapal

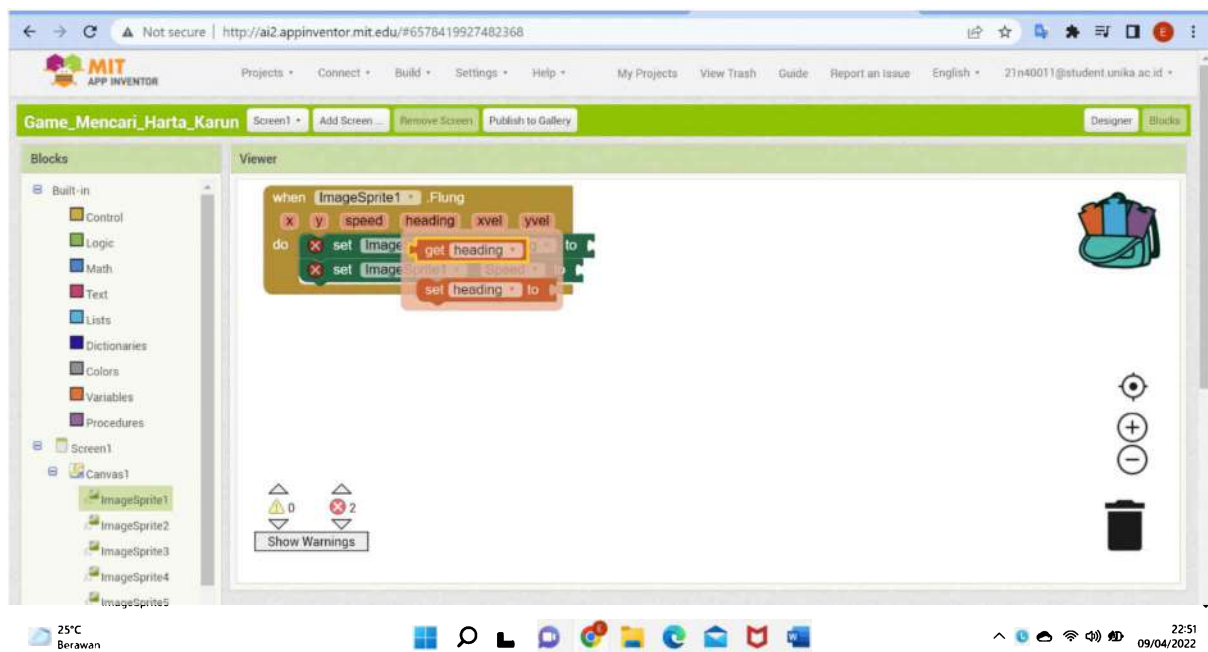


Gambar 9.13. Tampilan halaman *blocks*

Lalu pasangkan kedua *block* di atas ke dalam *handle when ImageSprite1.Flung*.

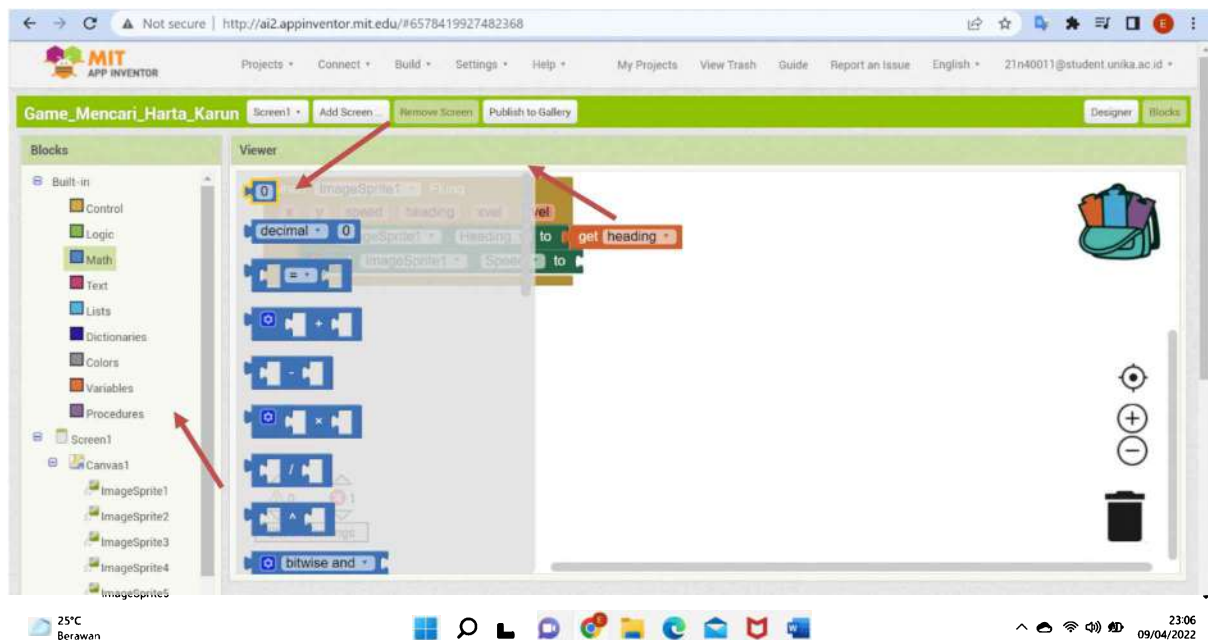
13. Arahkan *mouse* (tanpa mengklik) di atas parameter “*heading*” dari *handle when ImageSprite1.Flung*, seketika akan muncul menu *pop up* yang berisi *block* *get heading* dan *set speed to*, kita klik dan seret *get heading* lalu pasangkan ke

ujung set *ImageSprite1.Heading to*.



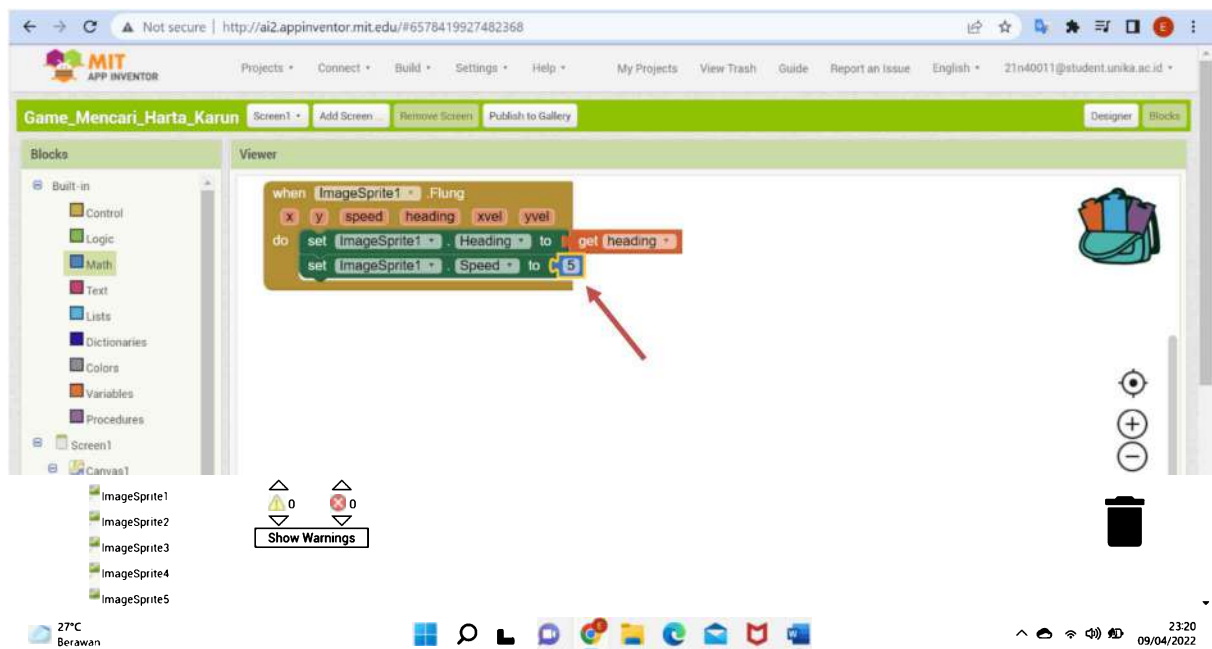
Gambar 9.14. Tampilan halaman *blocks*

14. Klik *Math* dan cari *block* perintah 0 lalu pasangkan ke set *ImageSprite1.Speed to*.



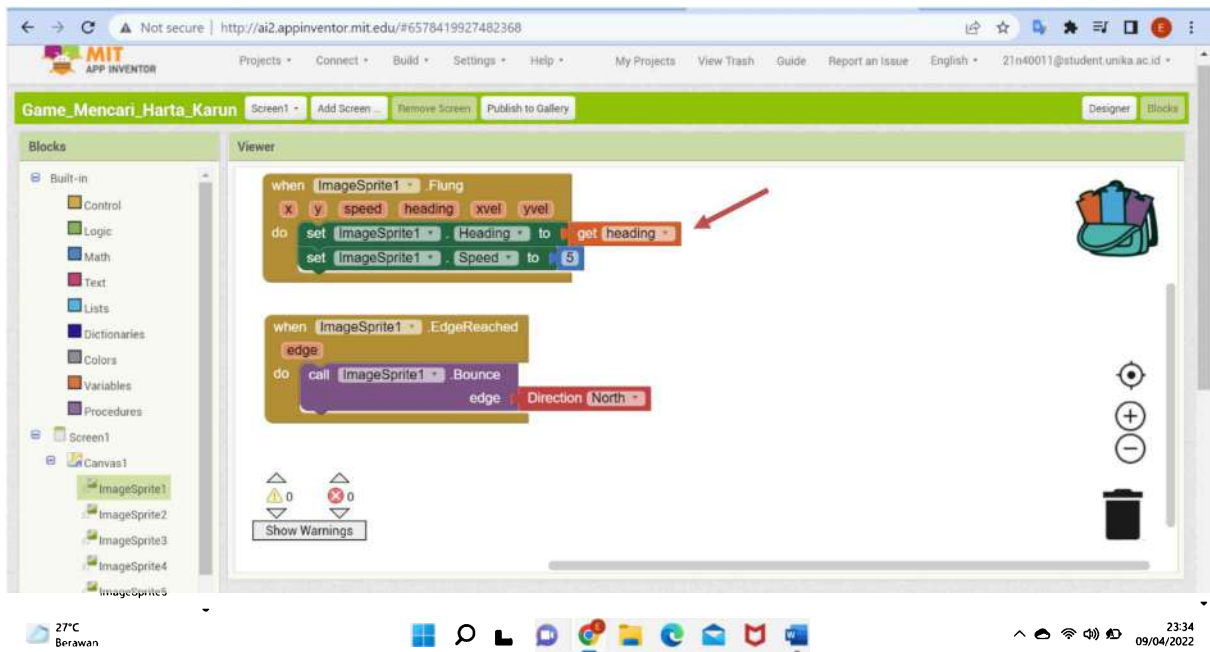
Gambar 9.15. Tampilan halaman *blocks*

15. Ubah *block* perintah yang tadi dari 0 menjadi 5.



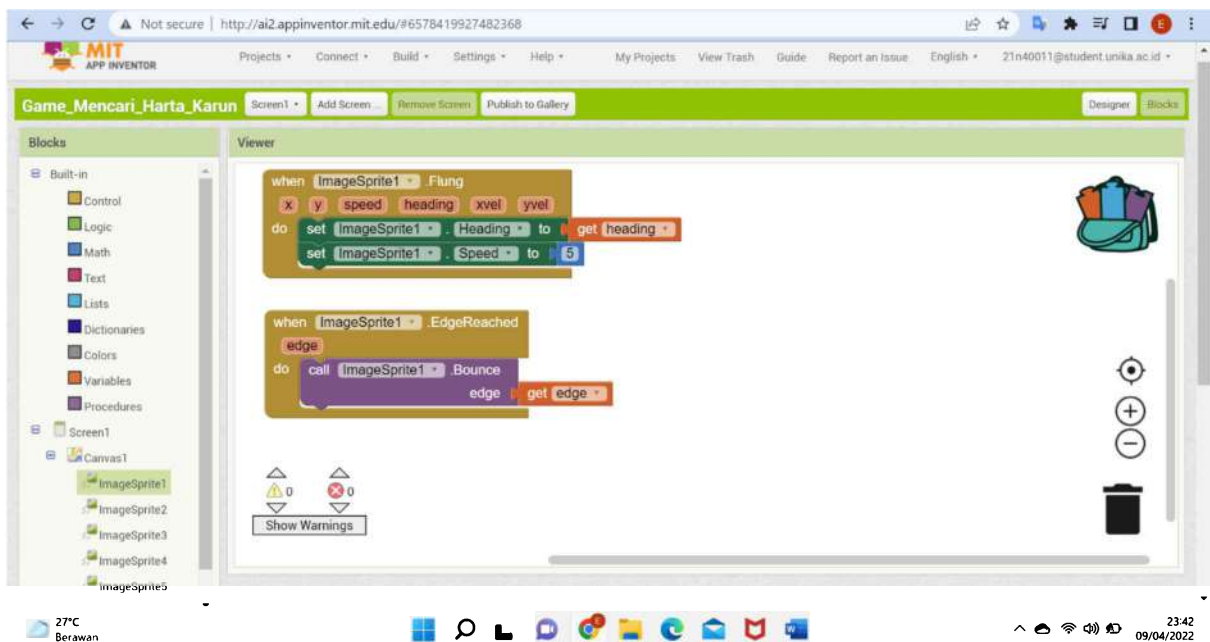
Gambar 9.16. Tampilan halaman *blocks*

16. Klik *ImageSprite1* kemudian cari *block* *when ImageSprite1.EdgeReached* dari daftar, seret dan lepaskan di area kerja. Klik kembali *ImageSprite1*, cari *block* *call ImageSprite1.Bounce* dari daftar, seret dan pasang pada *handle when ImageSprite1.EdgeReached*.



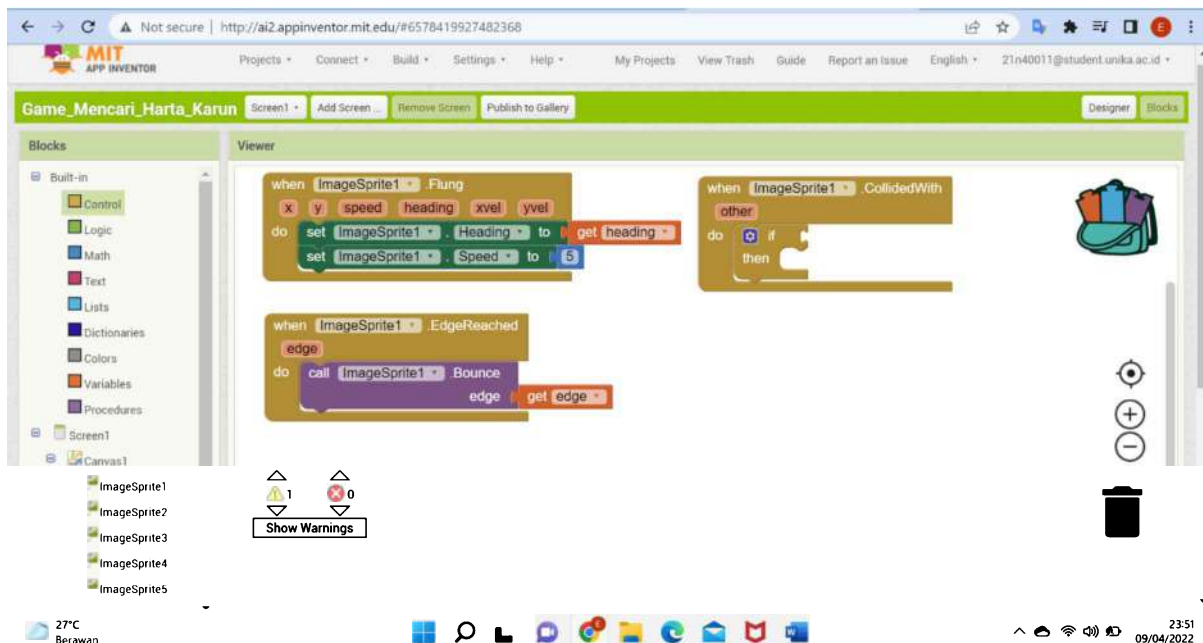
Gambar 9.17. Tampilan halaman *blocks*

17. Hapus *block Direction North*, lalu ganti dengan *block get edge* yang bisa kita ambil di parameter *edge* pada *handle when ImageSprite1.EdgeReached*



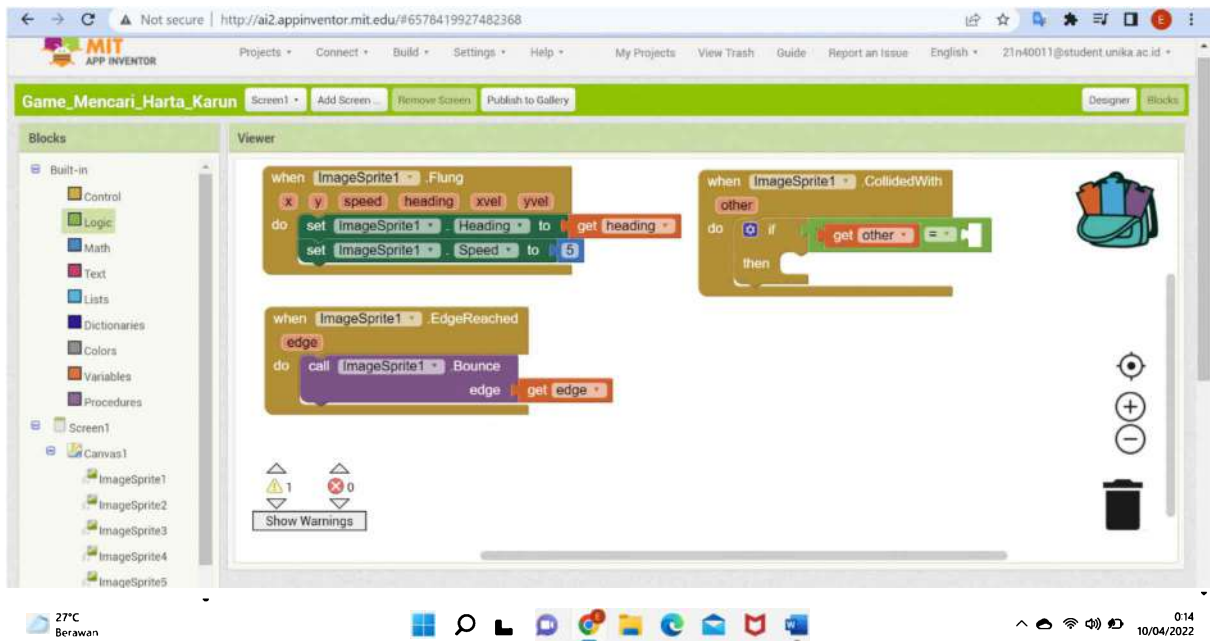
Gambar 9.18. Tampilan halaman *blocks*

18. Klik *ImageSprite1* kemudian cari *block when ImageSprite1.CollidedWith* dari daftar, seret dan lepaskan di area kerja. Klik *Control* lalu cari *block if then* dari daftar, seret dan pasangkan ke *handle when ImageSprite1.CollidedWith*.



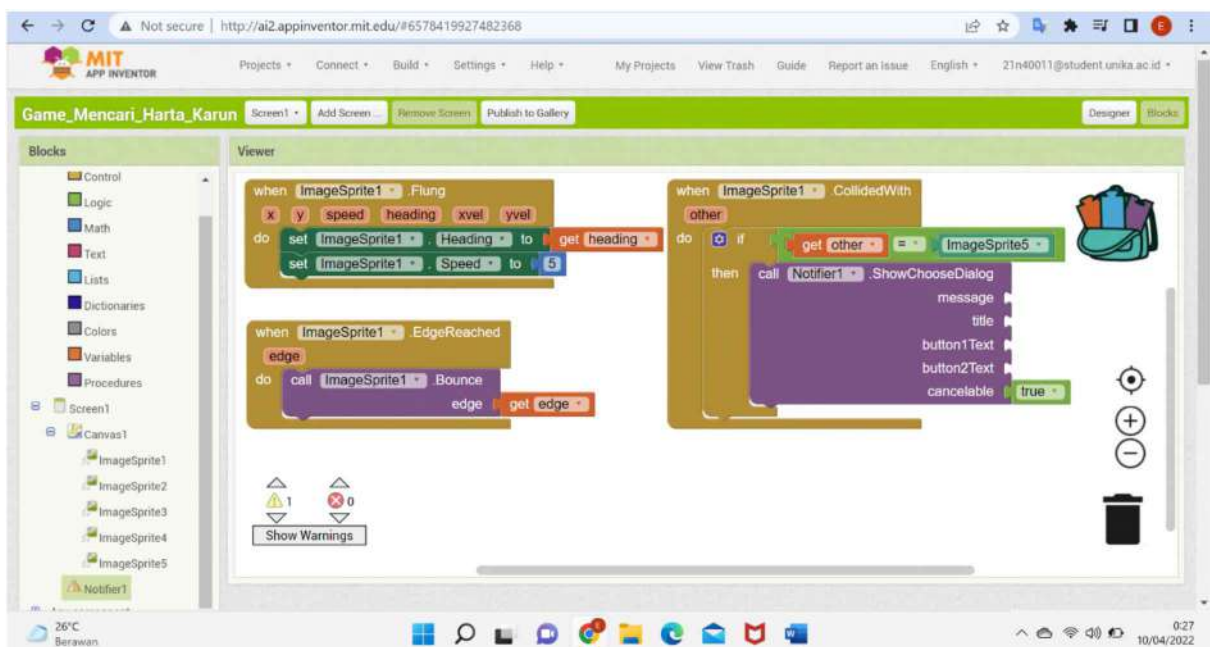
Gambar 9.19. Tampilan halaman *blocks*

19. Klik *Logic* lalu pilih “=” kemudian pasangkan ke *block if then*. Tambahkan *block get other* di parameter *other* pada *handle when ImageSprite1.CollidedWith* kemudian pasangkan seperti gambar di bawah ini.



Gambar 9.20. Tampilan halaman *blocks*

20. Klik *ImageSprite5* atau *ImageSprite* harta karun lalu pasangkan seperti di gambar. Selanjutnya klik *Notifier1*, pilih *call Notifier1.ShowChooseDialog* dan pasangkan ke *block* then.



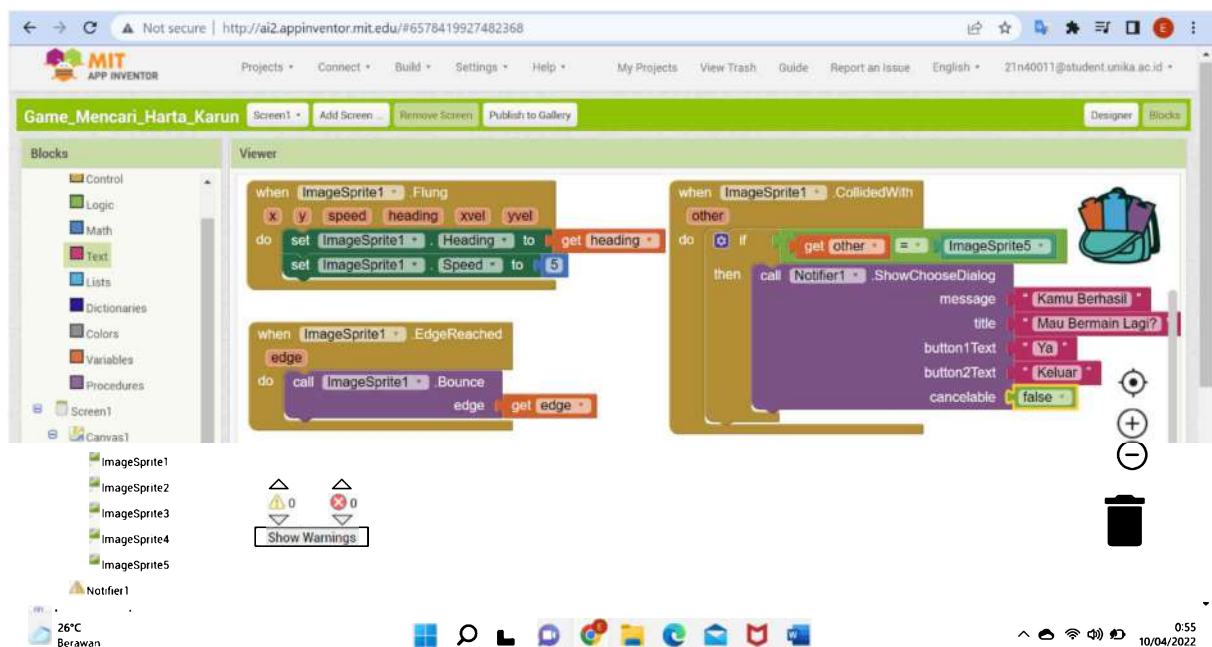
Gambar 9.21. Tampilan halaman *blocks*

21. Klik Text lalu pilih block text string kemudian pasang ke message, title, button1Text, dan button2Text.



Gambar 9.22. Tampilan halaman *blocks*

22. Isikan *block text string* yang kosong tadi dengan kalimat seperti gambar di bawah ini. Selanjutnya ubah *cancelable* dari *true* menjadi *false*.



Gambar 9.23. Tampilan halaman *blocks*

23. Klik *ImageSprite1* atau *ImageSprite* kapal lalu pilih *ImageSprite1.MoveTo* kemudian pasangkan ke *block* when *ImageSprite1.CollidedWith*. Setelah itu klik *Math*, pilih *block* perintah 0 lalu pasangkan ke bagian x dan y.



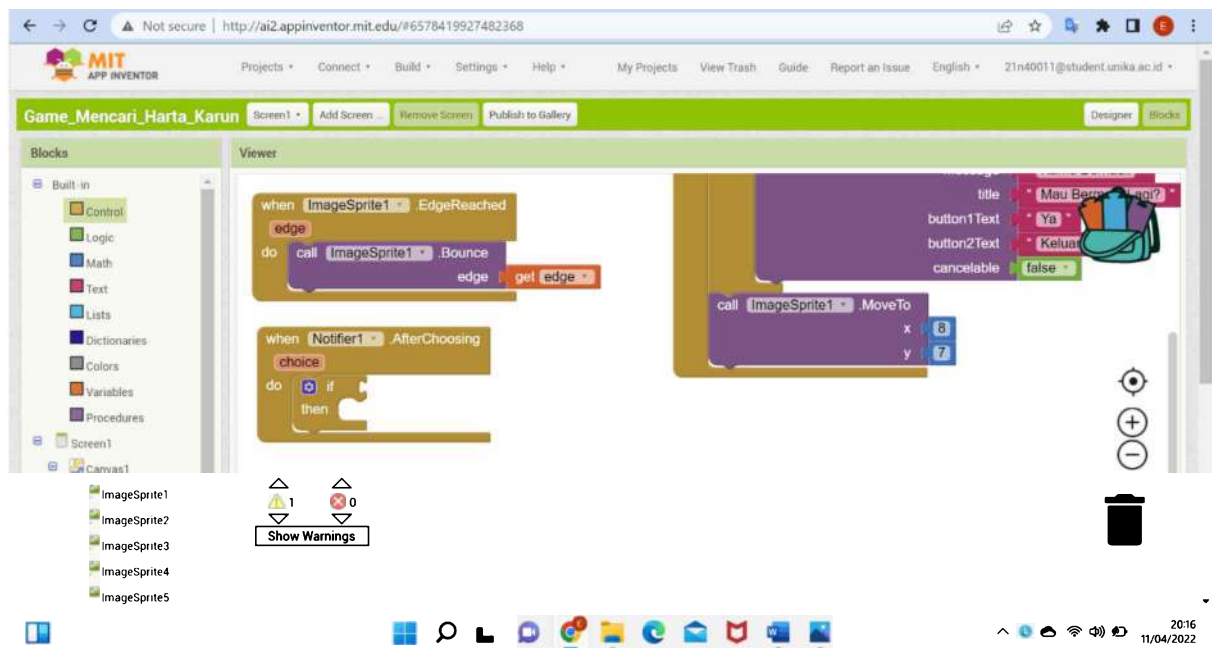
Gambar 9.24. Tampilan halaman *blocks*

24. Ubah *block* perintah 0 tadi dari 0 menjadi 8 untuk x dan 7 untuk y.



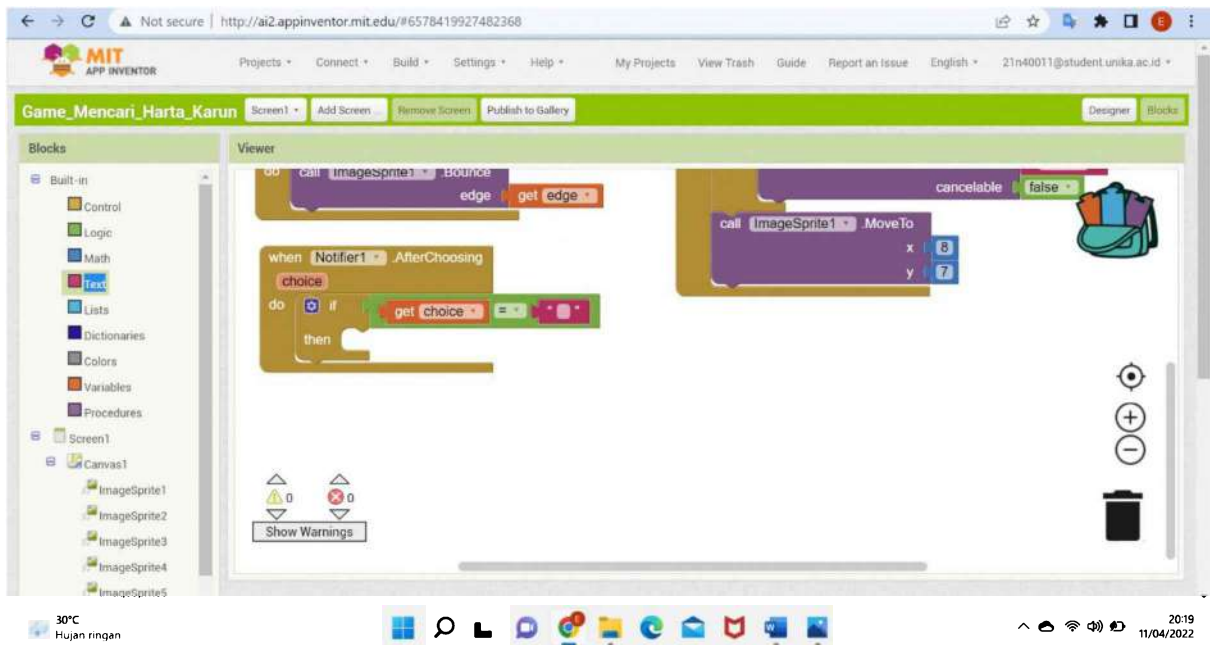
Gambar 9.25. Tampilan halaman *blocks*

25. Klik *Notifier1* lalu pilih *when Notifier1.AfterChoosing*, lalu klik *Control*, masukkan *block* perintah *if then* dan gabungkan seperti gambar dibawah ini.



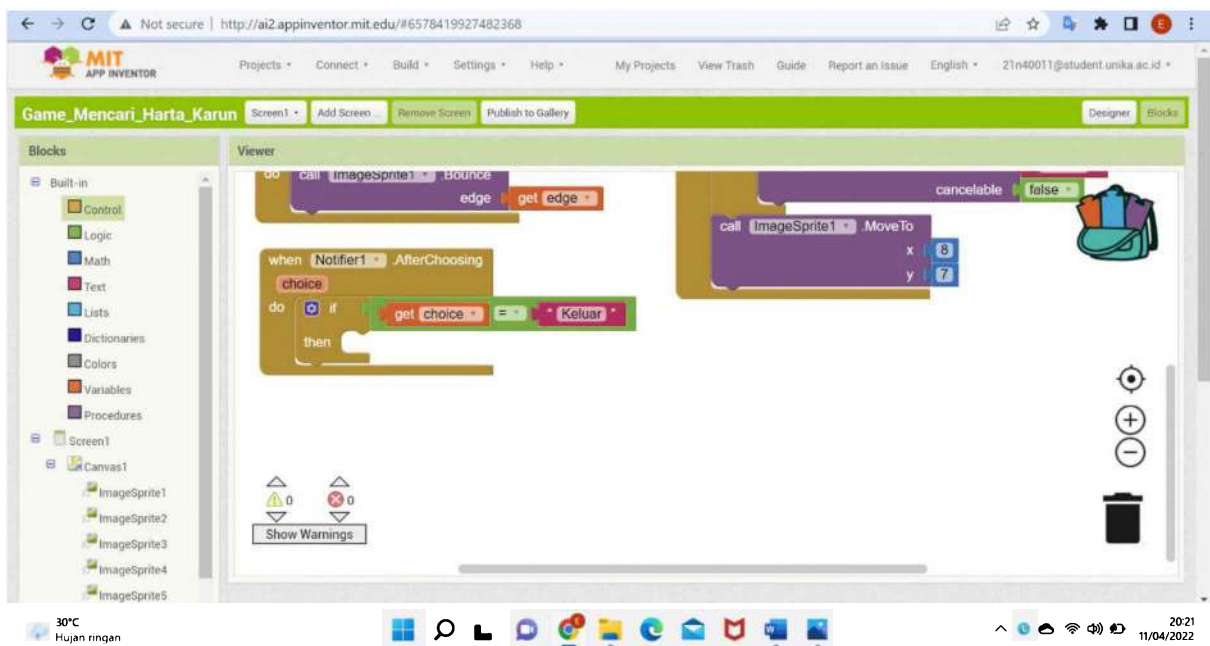
Gambar 9.26. Tampilan halaman *blocks*

26. Klik *Logic* lalu pilih “=” kemudian pasangkan ke *block if then*. Tambahkan *block get other* di parameter *other* pada *handle* when *Notifier1.AfterChoosing* dan tambahkan *text string* dan gabungkan seperti ini.



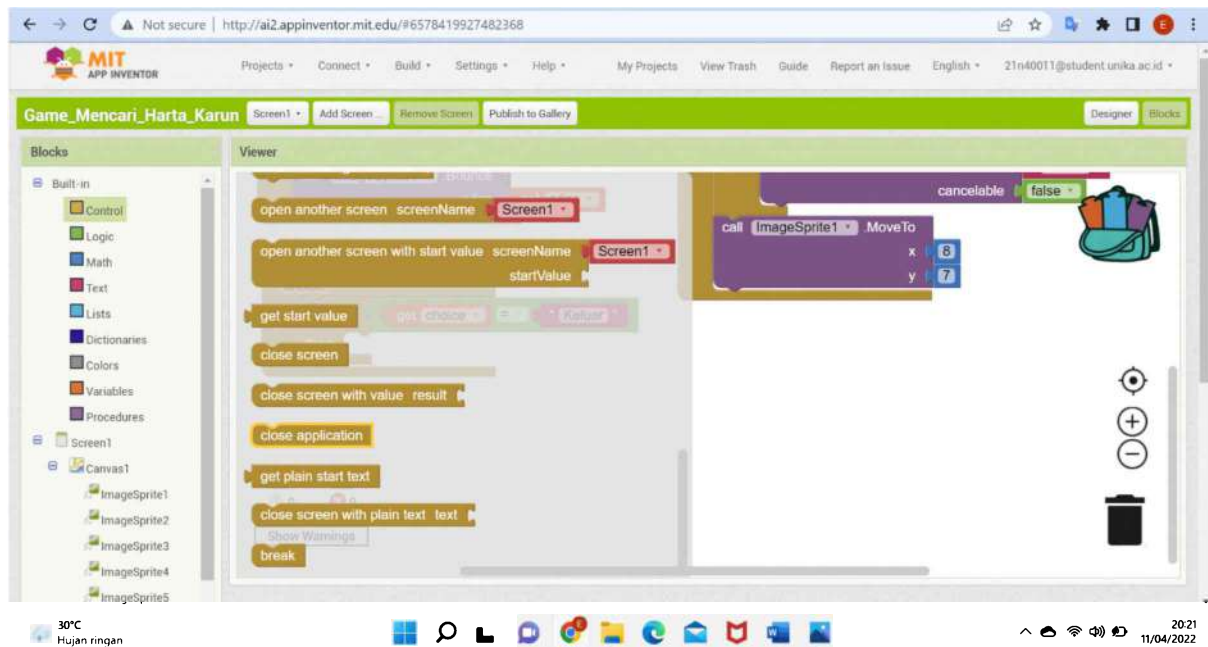
Gambar 9.27. Tampilan halaman *blocks*

27. Isi text string yang kosong tadi dengan kata “Keluar”.



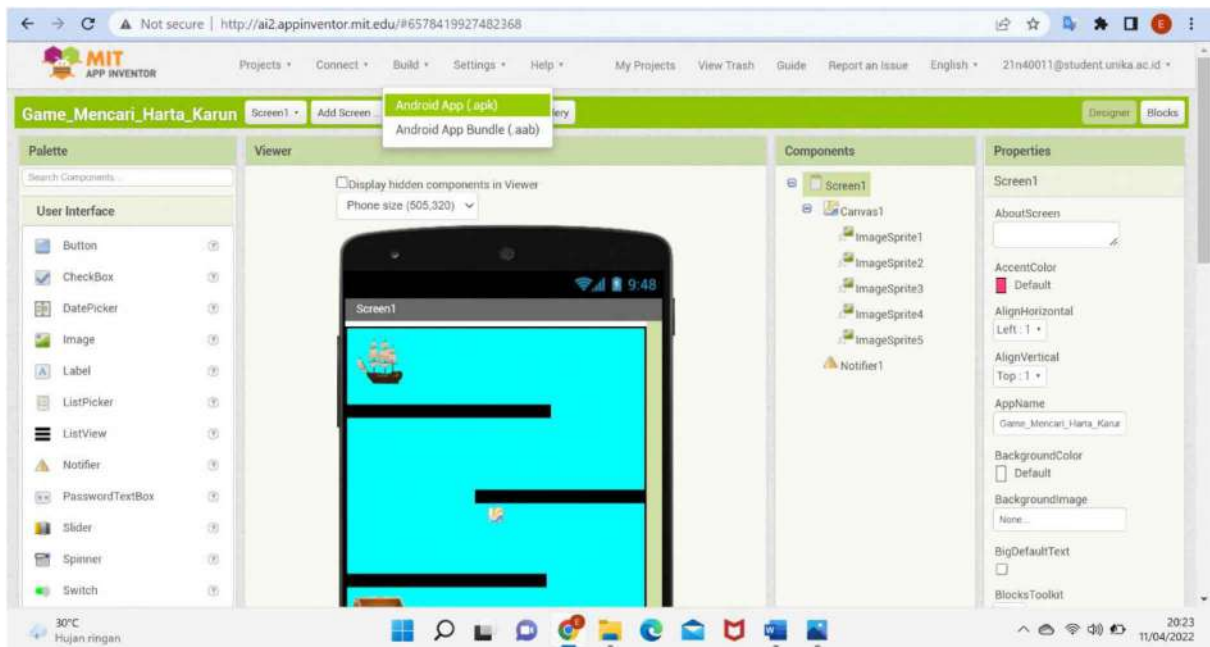
Gambar 9.28. Tampilan halaman *blocks*

28. Klik *Control* lalu masukkan *block close application* untuk dapat melakukan perintah keluar dari aplikasi. Kemudian gabungkan ke bagian *when*.



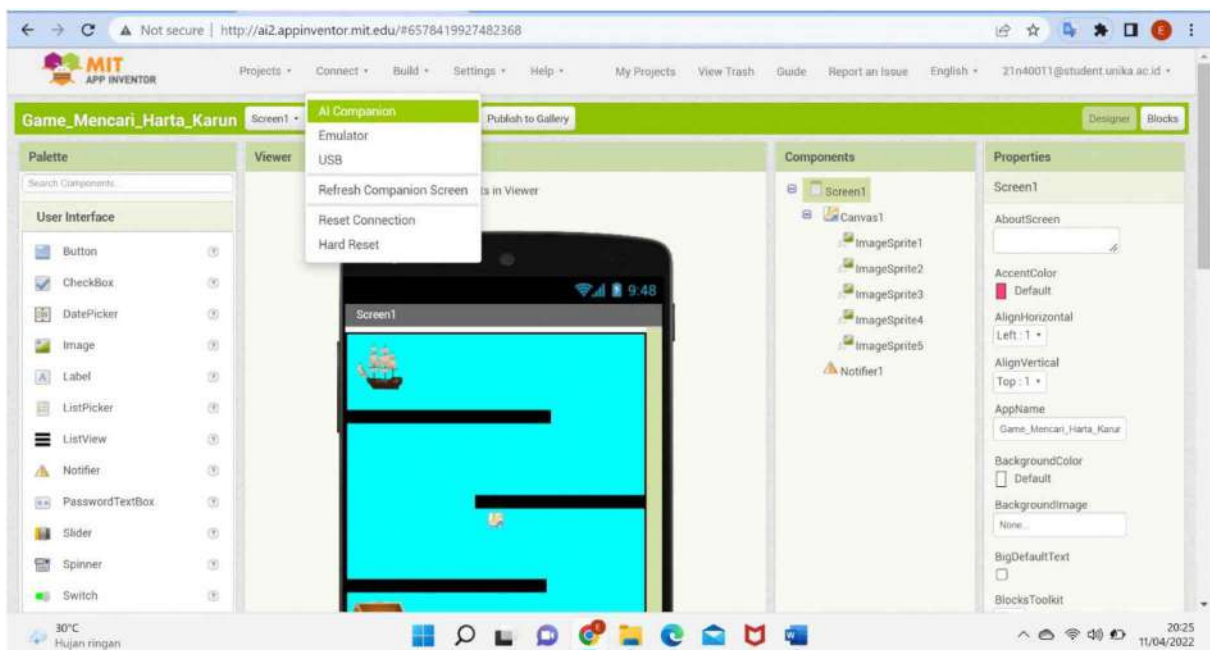
Gambar 9.29. Tampilan halaman *blocks*

29. Dengan begitu kita sudah menyelesaikan *block* dan desainnya, untuk dapat memainkannya kita bisa pilih menu *Build*, Pilih *Android App (Save.apk to my computer)*.



Gambar 9.30. Tampilan untuk *build apk*

30. Atau kita bisa juga menggunakan *emulator* MIT AI2 Companion yang bisa kita *download* di Play Store.



Gambar 9.31. Tampilan *connect emulator*

Link:

<https://drive.google.com/drive/folders/1X8Ooz7IVkvexzK2m8k-WKtviF4F8Bv5?usp=sharing>

Tantangan Tebak Kata Dan Gambar

Benedictus Brian Renaldi Darma

Di zaman yang modern ini teknologi sangat maju pesat contohnya teknologi aplikasi *game*, Di era sekarang game sudah sangat berkembang pesat dan diterima di berbagai kalangan masyarakat. Keahlian dalam bidang game juga sangat berkembang, Salah satunya adalah *smartphone*. *Smartphone* dibuat untuk menggantikan fungsi *computer* dalam bentuk yang lebih kecil. Salah satu contohnya adalah android yang merupakan sistem operasi yang bersifat *open source*, dimana pengembang aplikasi dapat membangun aplikasi berkualitas dan membuat segalanya lebih mudah bagi orang lain secara gratis.

Dalam dunia pendidikan, teknologi memiliki peranan tersendiri dalam proses belajar mengajar. Salah satunya adalah pembelajaran melalui media *game* tebak gambar dan menyusun kata. Pada saat ini dalam pengenalan huruf, bilangan, dan warna membutuhkan media sebagai sarana pembelajaran. Anak yang kurang memahami materi dari guru kemungkinan disebabkan oleh kurangnya alat peraga dan metode yang digunakan masih kurang tepat. Pada dasarnya gaya belajar merupakan metode terbaik yang memungkinkan dalam mengumpulkan dan menggunakan pengetahuan secara spesifik. Dalam buku ini dilakukan pengembangan *game* yang berupa pengenalan huruf, gambar dan bilangan dengan mengimplementasikan ke dalam *game*.

Tujuan dari pembuatan buku ini adalah untuk memudahkan anak-anak untuk menyusun kata dan bilangan secara interaktif agar dapat mengenalkan huruf dan bilangan. Aplikasi ini secara tidak langsung untuk menciptakan metode pembelajaran baru bagi anak-anak agar lebih mudah memahami huruf dan kata. Karena salah satu media yang dapat digunakan saat ini adalah media *game* sebagai media strategi pendidikan untuk menarik perhatian anak-anak agar lebih antusias *game* yang dinamakan *game* TANTANGAN TEBAK KATA. *Game* ini menggunakan kombinasi dari teka teki dan pendidikan. Dengan metode ini anak-anak akan sangat tertarik. *Game* edukasi dapat mendukung proses pembelajaran pendidikan. *Game* edukasi dapat diterapkan dengan karakteristik anak-anak yang suka bermain dan belajar, serta imajinasi mereka.

Bab Isi

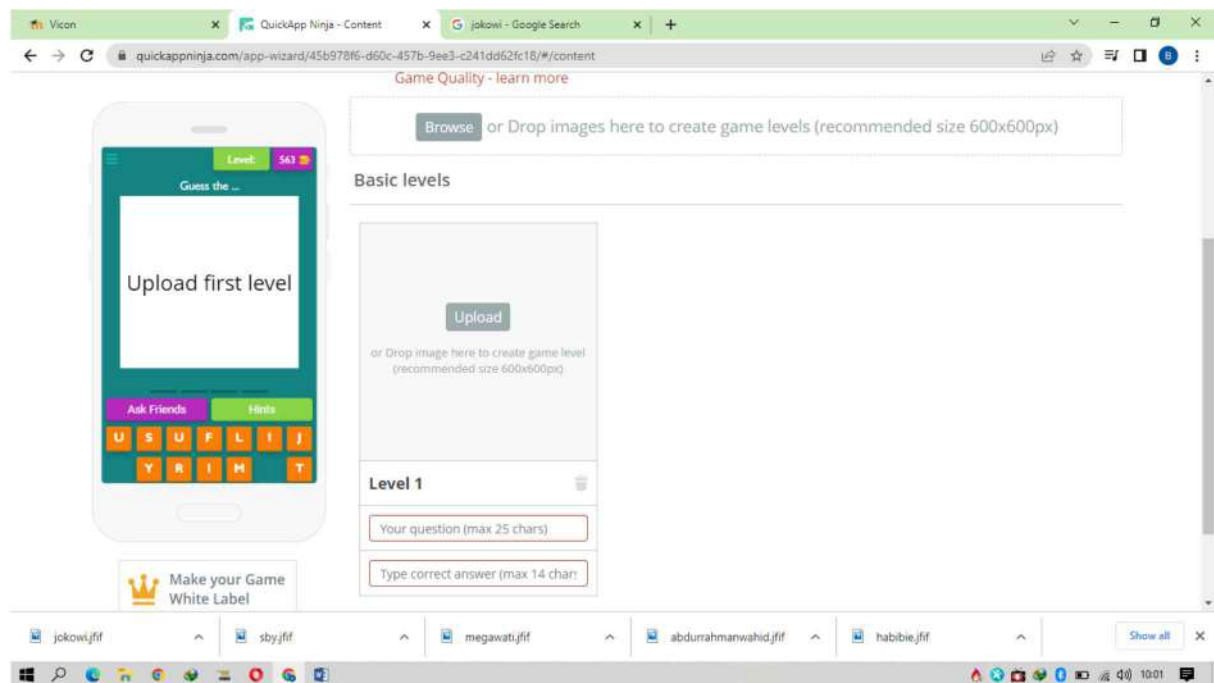
Inisiasi

Inisiasi merupakan proses guna menentukan konsep *game* yang akan dibuat nantinya. Biasanya proses penentuan ini dimulai dari penentuan *platform* yang menjadi sasaran *game* ini dimainkan, alur permainan, dan *game engine* yang akan digunakan

Storyboard

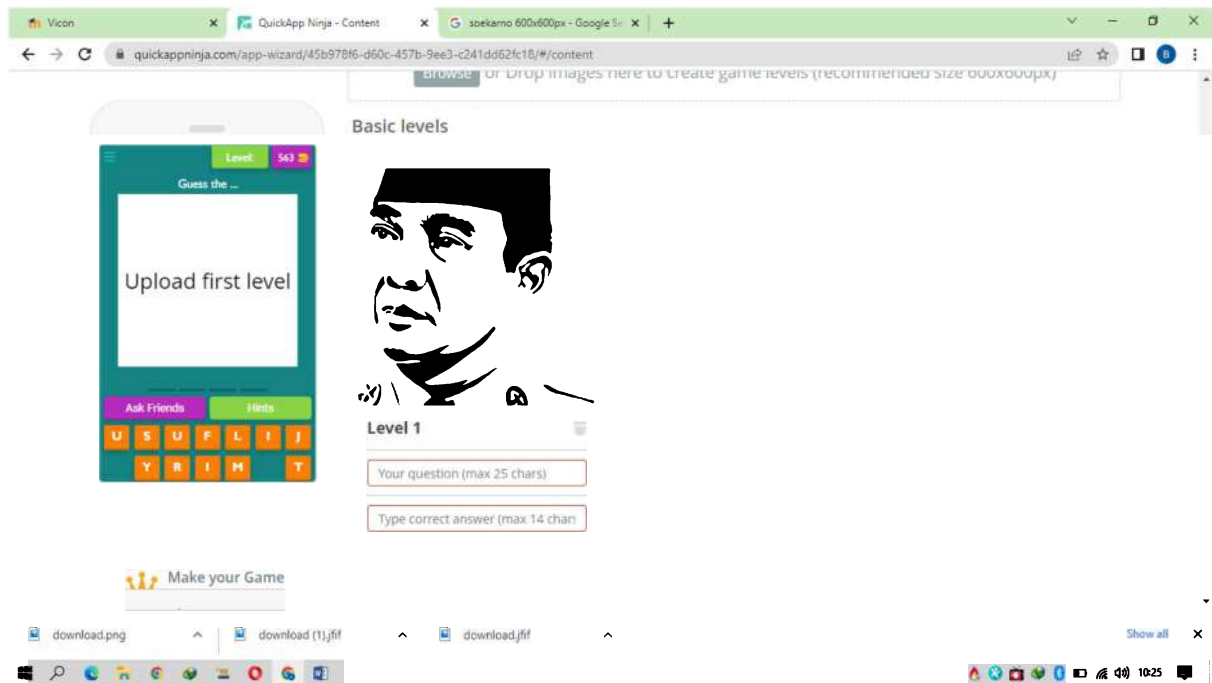
Dalam Storyboard merupakan gambaran mengenai *game* yang akan dibuat dalam bentuk sketsa kasar. Dalam sketsa ini mempermudah pemahaman mengenai gambaran *game* tersebut.

Pembuatan *Game*:



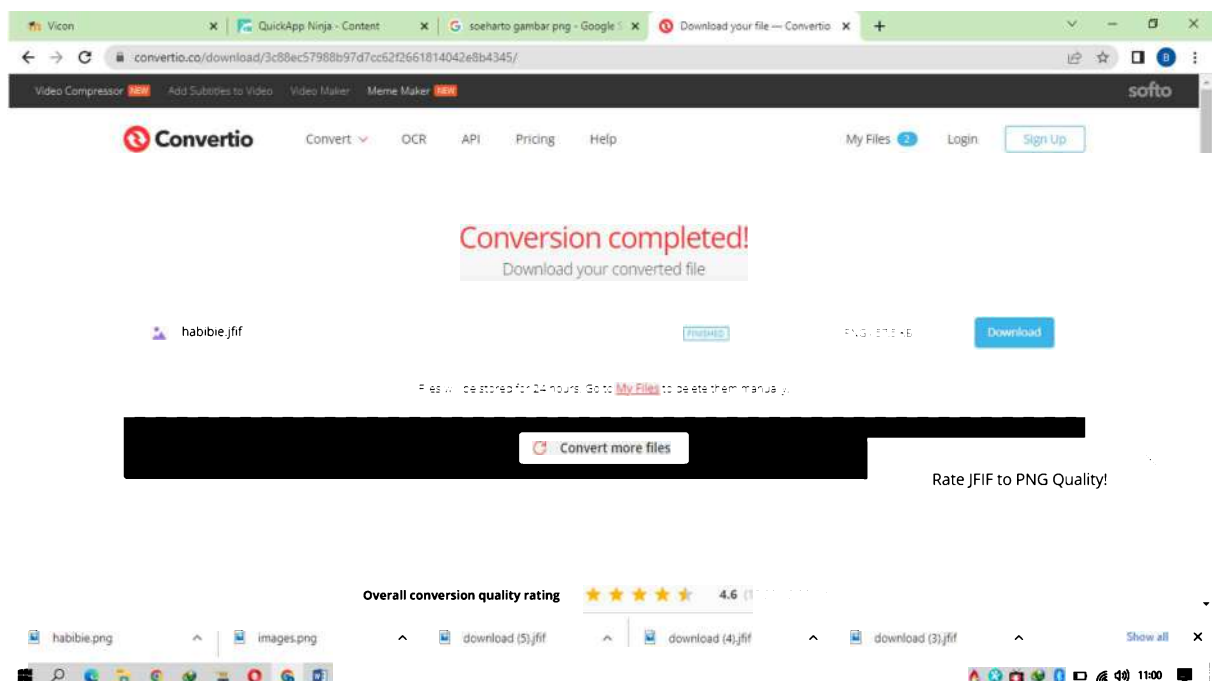
Gambar 10.1. Pembuatan *Game* di Quickapp Ninja

Saya menggunakan app Quickapp Ninja untuk pembuatan *game* tebak kata dan gambar ini di pembuatan *game* ini harus upload gambar untuk mengaplikasikannya maka dari itu saya *download* gambar-gambar presiden Indonesia

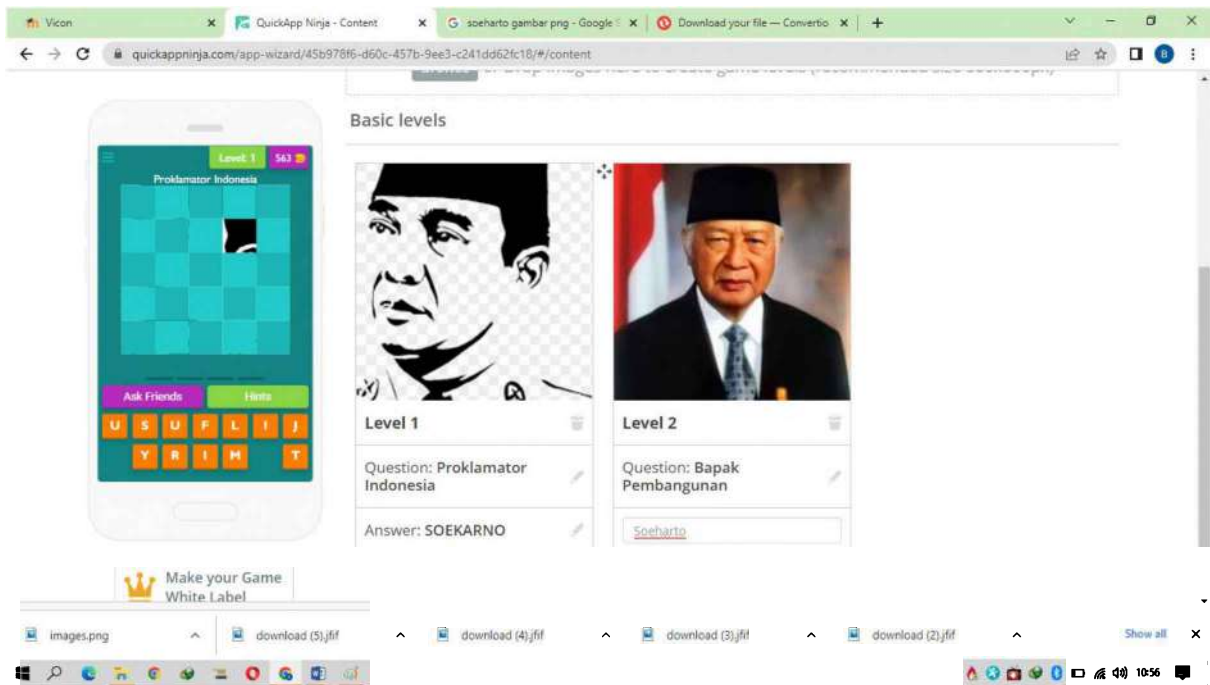


Gambar 10.2. *Upload* Gambar

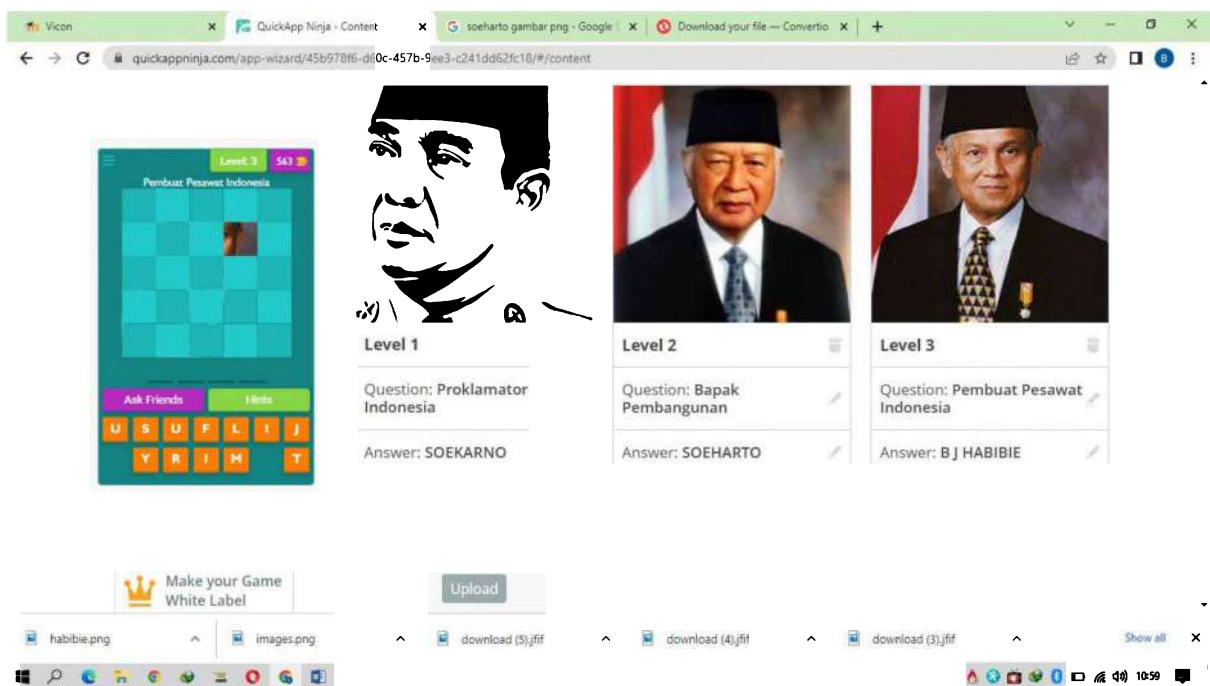
Akan tetapi file dari *download* masih jpg harus saya *convert* menjadi png



Gambar 10.3. Meng-*convert* Gambar Jpg ke Png

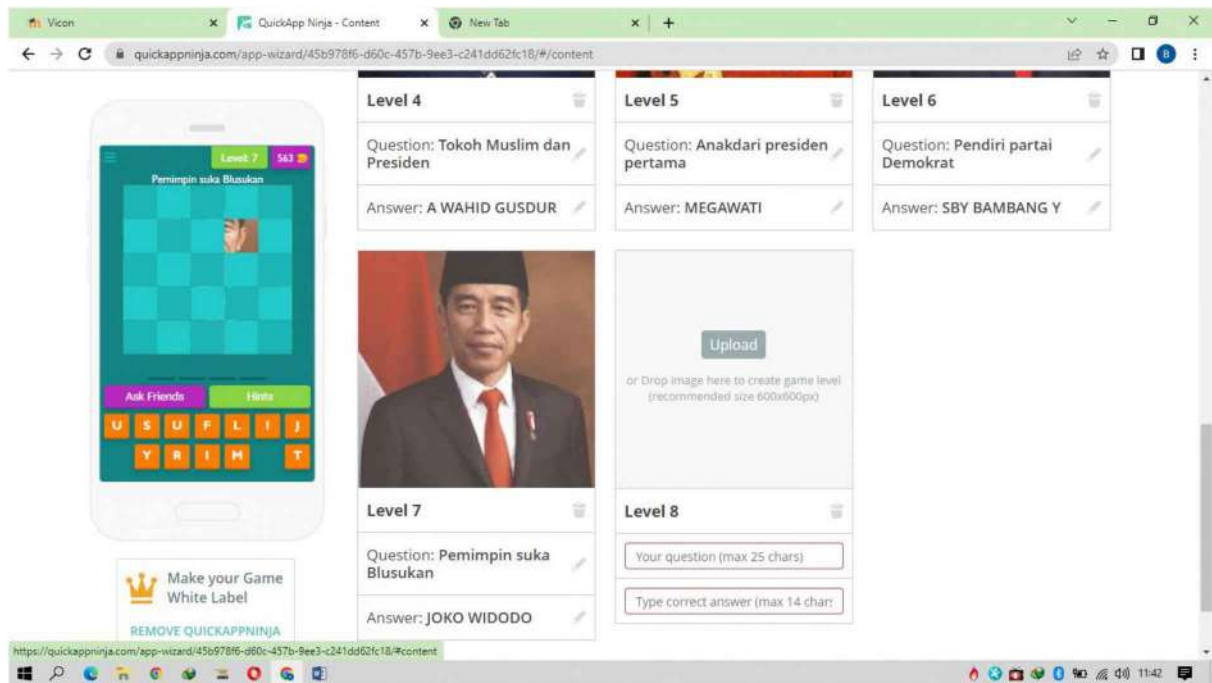


Gambar 10.4. Memberi Pertanyaan dan Jawaban



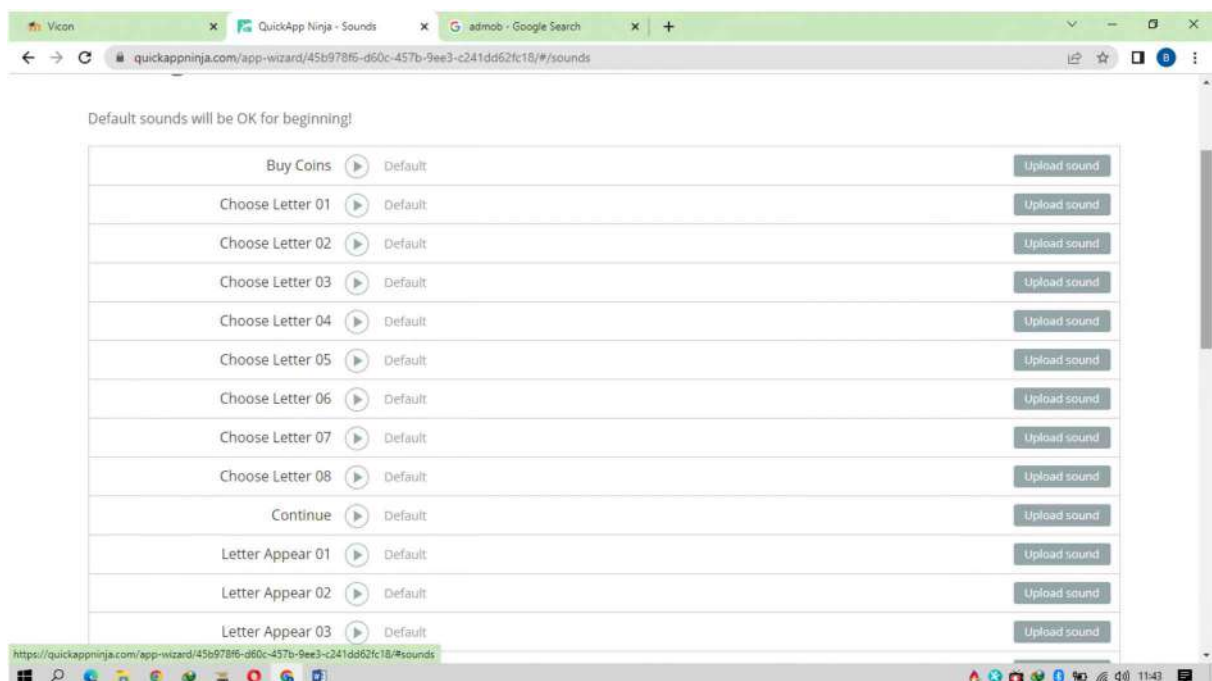
Gambar 10.5. Upload Pertanyaan dan Jawaban

Dari pembuatan *game* ini ada pertanyaan dan jawaban juga



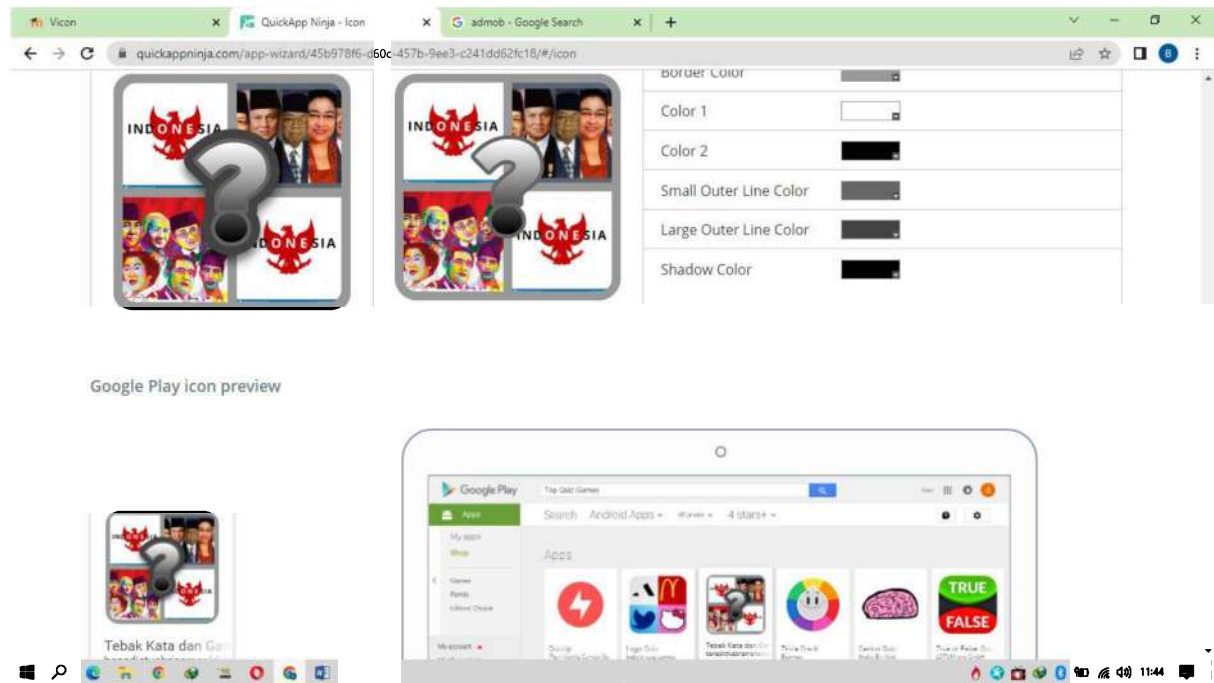
Gambar 10.6. Proses Pembuatan *Game*

Setelah itu saya juga mengedit beberapa *sound* untuk mengisi *game* tersebut



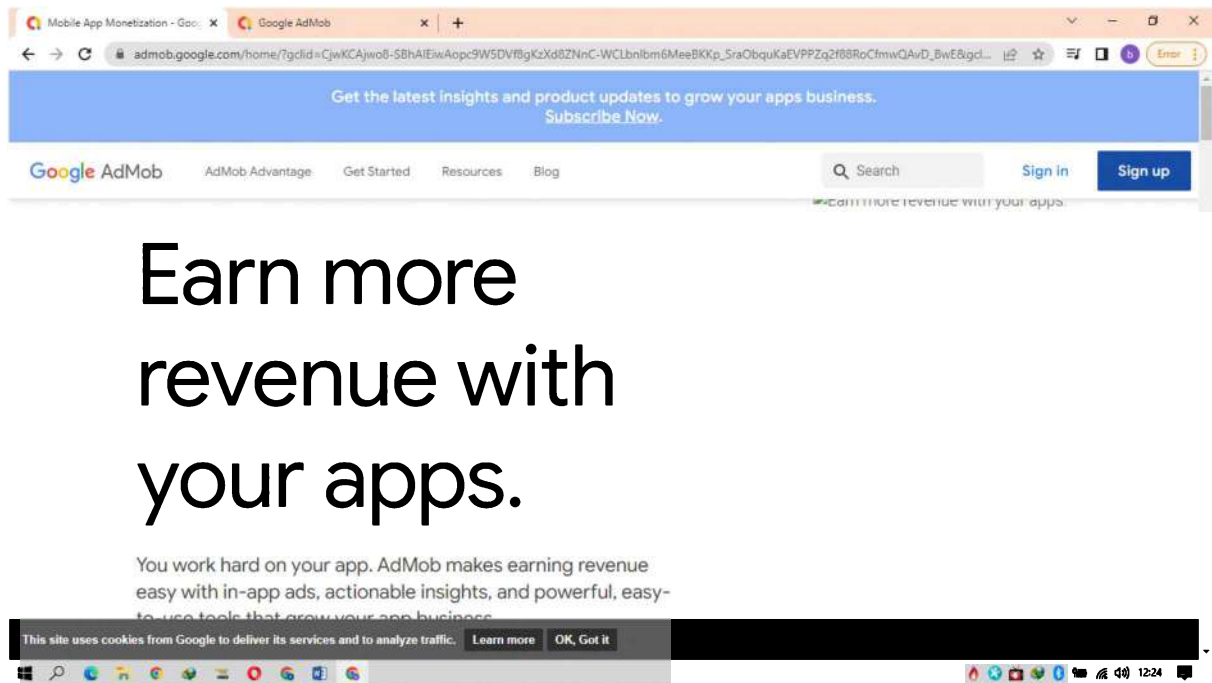
Gambar 10.7. Pengeditan *Sound Game*

Setelah mengedit *sound* saya juga mengedit *icon* untuk judul *game* dan *cover* dari *game* saya ini



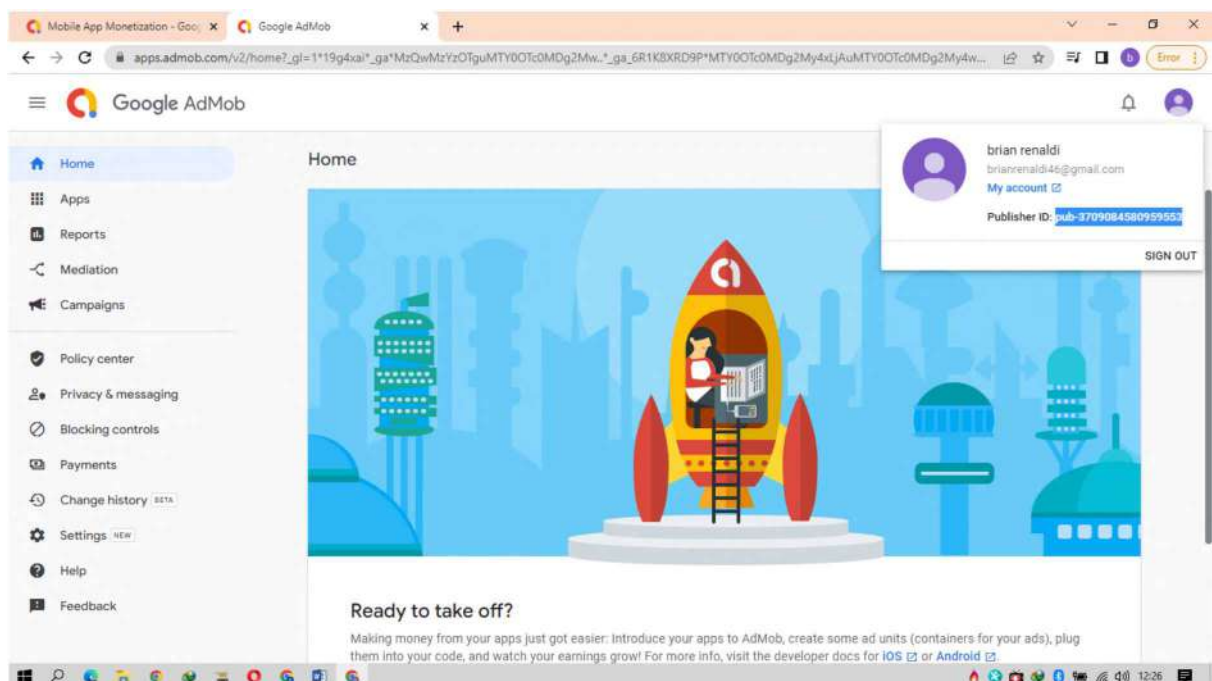
Gambar 10.8. Pengeditan *Icon Cover Game*

Langkah selanjutnya membuat akun Admob untuk melengkapi data dan mendapatkan keuntungan dari pembuatan *game* saya ini

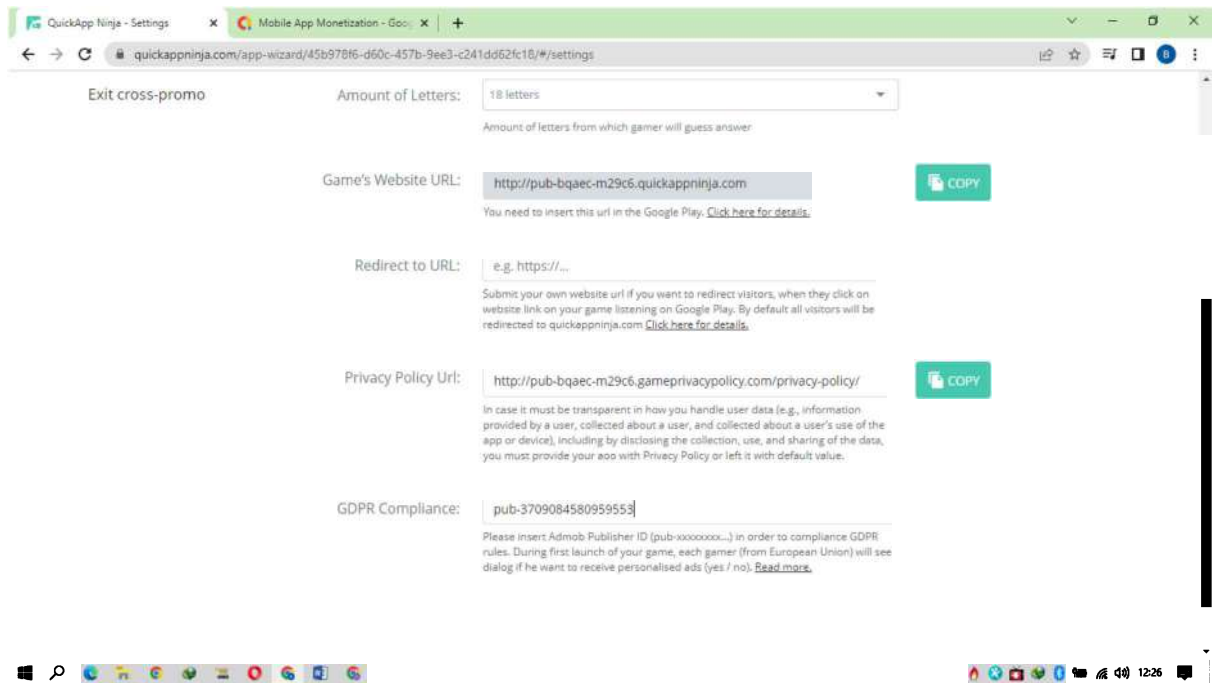


Gambar 10.9. Pembuatan Akun Admob

Setelah akun Admob jadi saya meng-copy *id* Admob saya ini untuk dimasukan ke app Quickapp Ninja

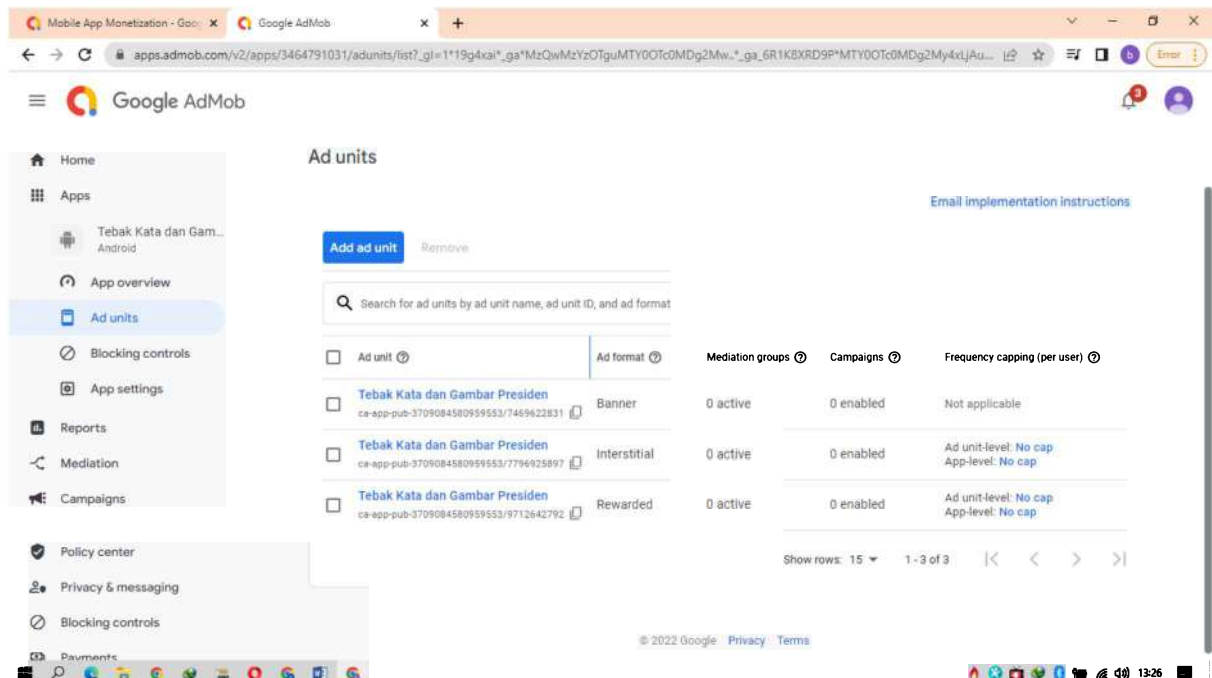


Gambar 10.10. Meng-copy *Id* Akun Admob



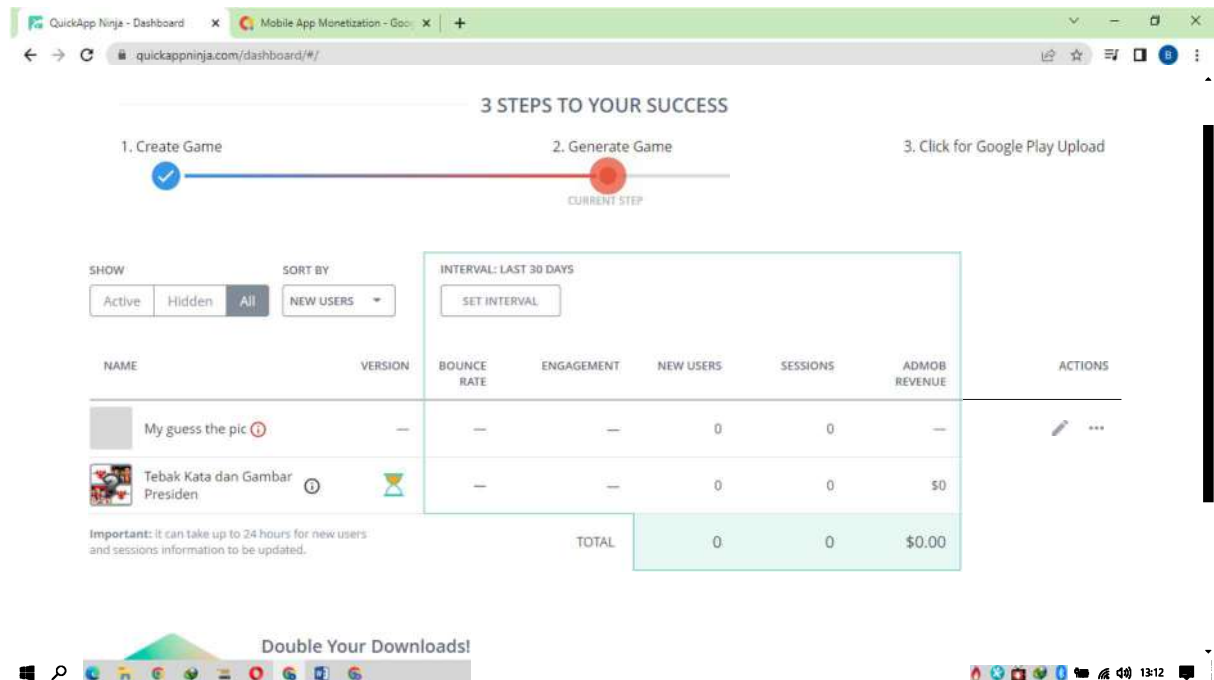
Gambar 10.11. Menyalin *Id* Akun Admob Menuju Quickapp Ninja

Setelah itu saya membuat pembuatan iklan di akun Admob



Gambar 10.12. Pembuatan Iklan di Akun Admob

Setelah semua langkah terpenuhi atau selesai langkah selanjutnya *generate game* saya agar bisa di *download* dan dimainkan *game* saya ini



Gambar 10.13. *Generate Game* Pengunduhan

Link Game:

https://drive.google.com/file/d/15GbkngWQ1BbDcFUW5Z_j-_jkbSDWY8NL/view?usp=sharing

Dengan ini saya membuat buku tentang Tebak Kata agar dapat melatih dalam belajar dan bermain *game*, Karena anak anak sangat antusias apabila bermain *game* maka dari itu saya mempunyai ide anak anak bermain *game* sekaligus belajar dengan cara ini dapat mengasah otak anak anak secara tidak langsung.

Halaman pembuka terdapat 3 tombol utama yaitu sebagai berikut:

1. *Button* pengaturan musik, berfungsi sebagai menyalakan musik dari aplikasi dan mematikan musik dari aplikasi.
2. *Button Start*, berfungsi sebagai ketika pengguna melakukan aktivitas klik pada tombol tersebut maka pengguna akan disuguhkan halaman materi.
3. *Button Exit*, yang berfungsi sebagai keluar dari aplikasi.

Apabila pada tombol *start* jika ditekan oleh pengguna, maka akan pindah halaman ke dalam menu aplikasi dan jika tombol silang ditekan akan keluar dari *game*.

Pada menu masing-masing materi terdapat beberapa *quiz*. Pengguna diminta untuk melengkapi kata yang terdapat pada gambar yang muncul pada *quiz*.

Menu petunjuk merupakan tampilan dari menu petunjuk. Ketika masuk ke dalam menu petunjuk yang ditampilkan adalah cara bermain berupa video langkah-langkah permainan. Setelah siswa mengerti cara memainkannya siswa dipandu untuk menekan tombol *back* pada halaman. Lalu halaman akan berpindah ke menu.

Pengujian *black box* dilakukan guna mengetahui apakah pada perangkat lunak telah berjalan sesuai dengan yang diharapkan sewaktu proses pengembangan atau tidak. Pengujian *black box* hanya dilakukan dengan memperhatikan proses *input*

serta *output* ketika perangkat lunak dijalankan. Pada pengujian ini penulis selaku pembuat perangkat ini yang melakukan pengujian dengan memasang aplikasi *game* pembelajaran ini pada ponsel yang penulis miliki.

Anda pasti tahu *game* Tebak Gambar yang sempat booming pada tahun 2013 di berbagai *smartphone* dan telah dimainkan oleh lebih dari 10 juta orang di Indonesia. *Game* yang cukup sederhana ini ternyata mampu menarik banyak perhatian orang yang memainkannya lantaran konsep tebak gambarnya yang bisa membuat pemain memutar otak sampai pusing tujuh keliling. Orang-orang tertarik untuk menyelesaikan tantangan demi tantangan yang ada.

Dalam setiap *level* terdapat tingkat kesulitan tertentu. Semakin tinggi *level* maka kesulitan yang didapat akan semakin sulit. Banyak kosakata-kosakata yang ternyata jarang diketahui oleh banyak orang sehingga sulit menebak kata demi kata. Hingga sekarang ini, *game* ini masih banyak dimainkan oleh beberapa orang. Meskipun sudah ada *game-game* serupa, Tebak Gambar tetap menjadi *game* yang begitu diminati dari tahun ke tahun.

kemudian memutar otak untuk menciptakan sebuah permainan tebak gambar yang dapat mengasah otak seseorang. Membuat permainan asah otak memang tidaklah mudah karena *developer* harus lebih pintar dari pemain. Banyak inspirasi kosakata yang mereka cari. Hingga awal

peluncurannya pada tahun 2013, *game* ini berhasil menarik banyak orang untuk memainkannya.

Sebenarnya bukan dari segi tipe permainannya, namun karena lebih merasa tertantang untuk menyelesaikan *level* demi *level* yang terlihat mudah namun ternyata sulit. Sekarang, Tebak Gambar sudah berkembang dengan ratusan *level* yang bisa dimainkan.

Semakin sulit, semakin banyak orang tertantang. Kunci-kunci jawaban pun banyak tersebar, dibuat oleh orang-orang yang berhasil menyelesaikan jawaban setiap *level*.

Tidak semua *game* harus memikat para pemainnya dari segi grafis yang bagus dan gambar-gambar yang memanjakan mata. Tebak Gambar memiliki desain gambar yang sederhana. Namun permainan kosakatanya selalu membuat penasaran setiap *level*.

Pemain akan disuguhkan potongan-potongan gambar yang akan disusun menjadi sebuah kata atau kalimat. Dengan beberapa petunjuk ada huruf-huruf yang dihilangkan.

Pada dasarnya, *game* ini merupakan permainan tebak kosakata. Hanya saja diberi petunjuk dengan gambar-gambar agar pemain dapat berpikir, gambar apakah ini.

Terkadang, setiap gambar memiliki lebih dari satu sebutan. Jika pemain salah menebak gambar padahal iya yakin benar, bisa jadi ada sebutan lain pada gambar tersebut. Seperti

gambar monyet yang ternyata kata yang dimaksud adalah
kera.

Setelah berhasil menebak gambar demi gambar, akan tercipta sebuah kosakata. Di akhir *level*, pemain disuruh menebak banyak gambar sehingga tersusun menjadi sebuah kalimat, ada waktu yang ditentukan sehingga pemain harus cepat. Tebak Gambar memiliki tombol bantuan yang dapat digunakan jika benar-benar tidak bisa memecahkan soal. Namun terbatas.

Konsep permainan dalam Tebak Gambar yang diusung oleh Lukis dan Irwanto ini bisa membuat seseorang mengasah otaknya dan menghafal beberapa istilah-istilah bahasa Indonesia yang masih asing untuk didengar. Di dalam *game* ini juga terdapat peribahasa dan beberapa perumpamaan dalam Bahasa Indonesia. Sehingga secara tidak langsung sangat cocok untuk membantu pembelajaran Bahasa Indonesia.

pemain diharuskan menyusun gambar demi gambar menjadi sebuah kosakata tertentu. Semakin panjang susunan gambar yang dibuat, maka akan semakin banyak poin yang didapatkan.

Permainan dapat dimulai dengan jumlah pemain minimal 2 sampai 4 orang. Setiap pemain dibagikan masing-masing 5 kartu, dengan menyisakan 4 kartu terbuka di tengah, dan sisa kartu diletakkan tertutup.

Pemain harus menyusun minimal dua buah kartu untuk merangkai sebuah kata atau kalimat. Jika tidak bisa menemukan kartu yang bisa dirangkai, bisa dengan mengambil yang di tengah. Pemain yang paling banyak menyusun kartu mendapat poin paling banyak.

Permainan Tebak Gambar yang dari tahun 2013 selalu menemani kita dikala waktu kosong ternyata memiliki perjalanan karir yang cukup panjang. Semua itu akan berkembang. Tebak Gambar sukses menjadi sebuah merek permainan yang kini dikenal oleh banyak orang di Indonesia. Sambil bermain melalui *smartphone*, pemain juga bisa mengajak teman-temannya bermain dengan kartu untuk menambah keseruan bersama.

Salah satu informasi yang dapat dikemas menjadi suatu media yang mampu memberikan pengalaman baru menggunakan perangkat *mobile* yaitu *game* edukasi yang dapat melatih daya pikir dan daya ingat serta mampu mengenalkan budaya dan potensi produk lokal ke masyarakat luas. *Game* edukasi tebak gambar dapat digunakan sebagai media informasi yang bersifat edukatif serta promotif. Dengan *game* edukasi tersebut pengguna tidak hanya bermain *game* namun dapat melatih daya ingat serta mendapatkan informasi mengenai budaya dan potensi produk asli daerah khususnya daerah Garut. Perancangan *game* edukasi tebak gambar ini menggunakan metodologi *digital game based learning - instructional design* yang terdiri dari 5 fase, yaitu *analysis*

phase, design phase, development phase, quality assurance kemudian *implementation and evaluation*. Hasil perancangan menunjukkan sebuah purwarupa *game* edukasi pengenalan budaya dan produk daerah melalui tebak gambar.

Game edukasi merupakan permainan yang dikemas untuk merangsang daya pikir termasuk meningkatkan konsentrasi dan memecahkan masalah. *Game* juga memiliki daya Tarik yang sangat baik karena mampu memberikan pengalaman yang baru sehingga informasi yang tersimpan di dalamnya mampu tersampaikan dan diingat oleh semua penggunanya, sebagian besar orang memiliki rasa ingin tahu yang tinggi mengenai segala hal yang berada di sekitarnya bahkan hal yang baru dilihatnya.

Game dapat digunakan sebagai sarana informasi, pendidikan, dokumentasi maupun hiburan, *game* juga sangat disarankan untuk media penyampaian informasi karena memiliki fitur yang lebih menarik dan interaktif. Masalah dalam suatu *game* dapat diselesaikan menggunakan strategi serta pemikiran yang cerdas. Tebak gambar merupakan masalah atau teka-teki yang diberikan sebagai hiburan sehingga diharuskan untuk menebak sebanyak-banyaknya kata, frasa, atau kalimat yang diisyaratkan ke dalam suatu gambar. Indonesia memiliki banyak budaya dan destinasi wisata serta produk daerah yang mampu menarik wisatawan asing, salah satunya di daerah Garut yang belum sepenuhnya dikenali oleh masyarakat luas. Garut memiliki banyak budaya dan potensi akan produk yang

perlu dikenalkan ke masyarakat luas, sehingga semua orang di seluruh Indonesia hingga mancanegara mengenal budaya dan produk asli Garut. Tujuan dari penelitian yang dilakukan ialah merancang *game* edukasi tebak gambar, sedangkan tahapan penelitian yang dilakukan berdasarkan tahapan-tahapan yang terdapat pada metodologi DGBL-ID, serta dalam melakukan langkah pengerjaan dari setiap tahapan yang dilakukan menghasilkan aktivitas-aktivitas, setiap fase memiliki aktivitas yang harus diselesaikan yaitu aktivitas yang disebutkan termasuk kegiatan utama untuk bagian instruksional dan bagian permainan.

Sebab kegiatan ini dilakukan atas dasar motivasi *internal* artinya kegiatan tersebut dilakukan bukan bermain merupakan suatu kegiatan yang sangat mengasyikkan atas dasar perintah maupun kehendak dari orang lain, tetapi karena keinginannya sendiri. Oleh karena itu, tak heran jika menghabiskan semua waktunya untuk bermain. Terkadang orang tua agak pusing dengan perilaku anaknya yang kalau sudah asyik bermain sulit untuk diajak berhenti sebentar walaupun hanya untuk sekedar makan atau tidur siang. Hal ini menunjukkan bahwa dalam kegiatan bermain anak sangat terlihat bahagia dan tidak kenal lelah atau selalu bersemangat seakan-akan energi mereka tak pernah habis. Karakteristik inilah yang membedakan antara dengan orang dewasa. Kegiatan bermain selalu dikaitkan dengan permainan. Secara umum permainan merupakan alat yang digunakan anak dalam kegiatan bermain sehingga kegiatan tersebut menjadi begitu

menarik dan berkesan bagi mereka. Namun, seiring perkembangan zaman kegiatan bermain dan permainan pun semakin berbeda sesuai dengan perkembangan teknologi yang semakin canggih. Jika orang terdahulu banyak menghabiskan waktu bermainnya bersama lain atau temannya dengan jenis permainan tradisional seperti bermain lompat tali, engklek, kelereng, pecah piring, bola kasti, petak umpet, congklak, bermain air di sungai, menjelajahi kebun atau ladang dan sebagainya. Tetapi anak yang tumbuh di zaman era globalisasi saat ini lebih senang menghabiskan waktu bermain mereka sendirian dengan menggunakan alat permainan yang disebut dengan *gadget*. Di dalam *gadget* mengenal *game online*.

Bermain sebagai pendekatan dalam pembelajaran anak usia dini hendaknya dilakukan dengan cara sederhana, menyenangkan, dan dengan media yang menarik. Salah satu permainan dalam mengembangkan kognitif adalah bermain tebak gambar.

Permainan ini begitu sederhana, namun jika guru kreatif dalam mengembangkannya maka akan menumbuhkan kemampuan anak dalam berbagai aspek perkembangan antara lain perkembangan kognitif anak.

Perkembangan kognitif yang timbul dari kemampuan berpikir dan memahami sesuatu semakin berkembang dengan baik jika terus belajar tentang kejadian atau peristiwa sekitar mainnya dan mampu berkomunikasi menurut usia dan

perkembangannya. Walaupun kemampuan kognitif ini dipengaruhi faktor bawaan, tidak menutup kemungkinan jika lingkungan ikut membantu atau mendukung maka kemampuan kognitif anak dapat berkembang dengan baik.

Dalam permainan tebak gambar kemampuan kognitif yang diharapkan salah satunya dapat mengasah kecerdasan anak dalam memahami hubungan objek yang dilihat sehingga imajinasi dan kemampuan berpikirnya tumbuh. Kegiatan bermain tebak gambar adalah kegiatan dimana guru mengajak menebak gambar yang ada dengan tujuan untuk menggali rasa ingin tahu dan kemampuan berpikir dalam menjabarkan sesuatu yang dilihat di sekitarnya.

1. Untuk mengembangkan kemampuan berpikir anak dalam memperoleh pengetahuan, kemampuan memecahkan masalah, mengembangkan kemampuan logika akan ruang dan waktu dan kemampuan berpikir teliti. Saat anak mampu berpikir logis diharapkan dapat memiliki pemahaman yang baik terhadap informasi (dalam hal ini gambar yang disajikan), mampu membandingkan dan membedakan. Pada akhirnya anak akan belajar memahami isi gambar yang ditampilkan
2. Anak dapat memahami berbagai konsep pengetahuan dari gambar yang ditampilkan. Konsep bentuk, warna dan ukuran, konsep matematika sederhana maupun konsep sains sederhana.

3. Mengetahui manfaat dan proses terjadinya sesuatu atau cara pembuatannya.
4. Melatih kemampuan dalam bertanya dan menjawab pertanyaan sederhana. Dari kemampuan ini tumbuh kemampuan penalaran dalam bahasa lisan sederhana.

Agar permainan tebak gambar menarik bagi anak usia dini, Kita hendaknya melakukan langkah-langkah, antara lain:

Pertama, siapkan gambar-gambar yang *full color* dan menarik perhatian anak sehingga anak mau diajak untuk bereksplorasi menemukan dan mengungkapkan ide dan menggali pengetahuan, rasa ingin tahu, dan menemukan hal baru dari gambar yang ditampilkan.

Kedua, menggali pemahaman dan wawasan anak tentang gambar yang disajikan dengan memancing pertanyaan-pertanyaan sederhana. Dari sini kadang kita menemukan jawaban atau ide yang tak terduga dari anak.

Ketiga, kembangkan wawasan anak. Artinya anak tidak saja memahami gambar yang disajikan, namun dapat diarahkan tentang warna, bentuk, ukuran atau menanyakan tentang proses sains sederhana, konsep matematika sederhana.

Keempat, dari permainan tebak gambar ini guru dapat memperkaya wawasan anak yang disesuaikan dengan situasi dan kondisi. Kreativitas guru atau orang tua dituntut agar anak semaksimal mungkin belajar dari kegiatan ini sesuai harapan. Dari kegiatan bermain tebak gambar pada akhirnya dapat

membantu mengembangkan daya pikir dalam mengenal, mengingat, dan memberikan kesimpulan.

Dari permasalahan tersebut penulis menggunakan model pembelajaran tebak kata agar peserta didik dapat memperhatikan dan lebih aktif dalam proses pembelajaran misalnya pada muatan pelajaran ilmu pengetahuan alam dan ilmu pengetahuan sosial yang banyak mengandung aspek pengetahuan.

Model pembelajaran tebak kata merupakan rumpun model pembelajaran *cooperative learning*, yaitu model pembelajaran yang menggunakan media kartu teka-teki yang berpasangan dengan kartu jawaban teka-teki. Permainan tebak kata dilaksanakan dengan cara menjodohkan kartu soal teka-teki dengan kartu jawaban yang tepat. Melalui permainan tebak kata selain anak-anak menjadi tertarik untuk belajar juga akan memudahkan anak-anak dalam menanamkan konsep pelajaran dalam ingatan. Selain itu peserta didik juga diarahkan untuk aktif yaitu memiliki kemampuan untuk bertanya dan mengemukakan gagasan. model pembelajaran tebak kata adalah model pembelajaran penyampaian materi ajar dengan menggunakan kata-kata singkat dalam bentuk permainan sehingga peserta didik dapat menerima pesan pembelajaran melalui kartu. Selain itu juga menjelaskan bahwa model pembelajaran tebak kata merupakan model pembelajaran yang menggunakan media kartu teka-teki yang berpasangan dengan kartu jawaban teka-teki.

1. Kelebihan model pembelajaran tebak kata

- a. Anak-anak akan mempunyai kekayaan bahasa.
- b. Sangat menarik sehingga setiap peserta didik ingin mencobanya.
- c. Peserta didik menjadi tertarik untuk belajar
- d. Memudahkan dalam menanamkan konsep pelajaran dalam ingatan peserta didik.

2. Kekurangan model pembelajaran tebak kata

- a. Memerlukan waktu yang lama sehingga materi sulit tersampaikan
- b. Bila anak-anak tidak menjawab dengan benar maka tidak semua siswa dapat maju karena waktu terbatas.

Salah satu model pembelajaran yang dapat menarik bagi anak-anak untuk mengikuti pembelajaran adalah model pembelajaran tebak kata di aplikasi kan ke dalam *game*. Model pembelajaran tebak kata ini merupakan model pembelajaran yang menggunakan media kartu teka-teki yang berpasangan dengan kartu jawaban teka-teki dan gambar. Melalui permainan tebak kata selain anak menjadi tertarik untuk mengikuti pembelajaran juga memudahkan untuk menanamkan konsep pelajaran dalam ingatan siswa. Dengan model pembelajaran ini diharapkan prestasi belajar anak-anak akan meningkat dan pembelajaran akan menjadi lebih

bermakna bagi anak-anak. Model pembelajaran tebak kata merupakan salah satu strategi pembelajaran yang diyakini mampu meningkatkan motivasi maupun prestasi anak-anak dalam belajar. Model pembelajaran ini dapat digunakan sebagai alternatif untuk menciptakan suasana/kondisi yang bervariasi dalam kegiatan belajar.

Model pembelajaran adalah suatu perencanaan atau suatu pola yang digunakan sebagai pedoman dalam merencanakan pembelajaran di kelas atau pembelajaran dalam tutorial. Bahwa model pembelajaran mengacu pada pendekatan pembelajaran yang akan digunakan, termasuk di dalamnya tujuan-tujuan pengajaran, tahap-tahap dalam kegiatan pembelajaran, lingkungan pembelajaran, dan pengelolaan kelas.

Daftar Pustaka

Download RPG Maker MV. (n.d.).
<https://www.rpgmakerweb.com/downloads>

Logo Unity. (n.d.).
[https://id.wikipedia.org/wiki/Unity_\(mesin_permainan\)](https://id.wikipedia.org/wiki/Unity_(mesin_permainan))

spesifikasi RPG Maker MV. (n.d.).
<https://www.rpgmakerweb.com/products/rpg-maker-mv>

Gamebrott.com. (n.d.). <https://gamebrott.com/10-game-engine-terbaik-di-dunia#:~:text=Cry Engine&text=Game yang berhasil di ciptakan,%3A Ghost Warrior 2%2C dll>.

Getective.com. (n.d.). <https://getective.com/game-engine-rpg-maker/>

Link Video. (n.d.). <https://youtu.be/MhW3UTZ9qGU>

portdesk.net. (n.d.). <https://portdesk.net/id/10-game-android-teratas-dibuat-dengan-unity>

RPG Maker Wikipedia. (n.d.).
https://en.wikipedia.org/wiki/RPG_Maker

Unity Wikipedia. (n.d.).
[https://id.wikipedia.org/wiki/Unity_\(mesin_permainan\)](https://id.wikipedia.org/wiki/Unity_(mesin_permainan))

Unreal Engine Wikipedia. (n.d.).
<https://www.brilio.net/creator/inilah-6-jenis-game-engine-yang-perlu-kamu-tahu-42da80.html>

Haydengames3v2. (2016). *RPG Maker MV - Blackjack Tutorial.*

<https://www.youtube.com/watch?v=kF8jYFzVT4A>

Biografi



Roberto Antonio Mozart Udju, yang biasa dipanggil Roberto, lahir di Wonogiri, 9 Mei 2003. Dia adalah mahasiswa Unika Soegijapranata, Angkatan 21, program studi Game Technology. Memiliki hobi bermain sepak bola dan game.

Perkenalkan nama lengkap saya Kevin Wijaya, saya lahir di kota Rembang, tanggal 20 Maret 2003. Saya berkuliah di kampus Unika Soegijapranata Semarang, mengambil program studi Game Technology. Hobi saya menonton film dan bermain game.



Penulis bernama lengkap Adinata Ekwana Januar Pratama yang lahir di Semarang pada 25 Januari 2002. Ia adalah anak pertama dari sepasang anak kembar. Sekarang ia sedang berkuliah di Unika Soegijapranata Semarang mengambil jurusan Game Technology.

Kristoforus Thoeirick Dimasaputra adalah seseorang yang lahir di kota Pangkalpinang pada tanggal 2 Oktober 2003, ia adalah anak kedua dari empat bersaudara.

Pada saat ini sedang menyelesaikan studinya di UNIKA Soegijapranata Semarang, masuk Fakultas Ilmu Komputer dan mengambil program studi Game Technology.

Ini adalah Tulisan Pertamanya.



Samuel Matthew Gunawan lahir di Semarang, 17 Agustus 2003. Dia adalah alumni dari SMA Karangturi Semarang. Sekarang adalah mahasiswa di Unika Soegijapranata jurusan Game Technology. Ini adalah tulisan pertamanya, semoga bermanfaat. Hobinya mendengarkan musik dan bermain game.

Marcello Jovan Christopher Kalbuadi lahir di Temanggung, 27 Maret 2003. Dia adalah alumni dari SMA Karangturi Semarang. Sekarang adalah mahasiswa di Unika Soegijapranata. Hobinya adalah bermain musik dan bermain game.



Penulis Bernama lengkap Renaldo Haryono Kusumo, yang biasa di panggil Aldo, lahir di kota Tegal, 4 September 2002. Sekarang dia adalah mahasiswa Game Technology dari Universitas Katolik Soegijapranata.



Penulis bernama lengkap Justin Hezekiel Ramli. Tempat tanggal lahir Semarang, 4 Mei 2003, anak kedua dari 5 bersaudara. Alumni SMA Kebon Dalem Semarang. Sekarang menjadi mahasiswa di UNIKA Soegijapranata Semarang. Ini merupakan karya tulis pertama

saya semoga dapat bermanfaat bagi pembaca.

Penulis Bernama Enrico Natanael Nugroho, lahir di Semarang pada tanggal 9 Mei 2003. Ia adalah anak kedua dari dua bersaudara. Sekarang ia berkuliah di Unika Soegijapranata Semarang mengambil jurusan E-Commerce.





Penulis bernama Benedictus Brian Renaldi Darmawan. Tempat lahir di Semarang 11 Juli 1999, dia adalah anak kedua dari dua bersaudara. Dia adalah alumni SMA Don Bosko Semarang dan sekarang melanjutkan studi di UNIKA SOEGIJAPRANATA Semarang.

Ini adalah pengalaman pertama kali nya membuat karya buku dan pembuatan game, Semoga bermanfaat.

Penulis adalah salah satu mahasiswa di Unika Soegijapranata angkatan 2017.

PROGRAMMER TANPA CODING

Buku ini membahas tentang game-game yang dibuat oleh mahasiswa Unika Soegijapranata tanpa melakukan coding menggunakan RPG Maker, MIT App Inventor, dan Quickapp Ninja. Buku ini juga dilengkapi dengan gambar agar pembaca dapat mengikuti tutorial dengan mudah.

Masih kurang?

Buku ini juga menyediakan link untuk menngunduh hasil game buatan para mahasiswa agar mempermudah proses belajar membuat game dan suatu program.