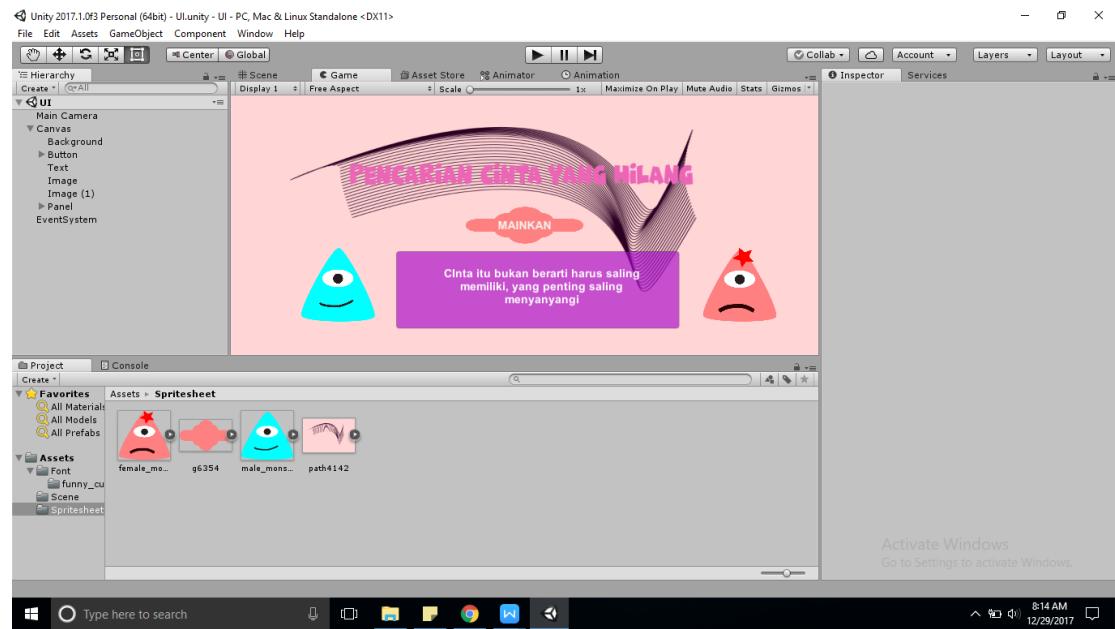


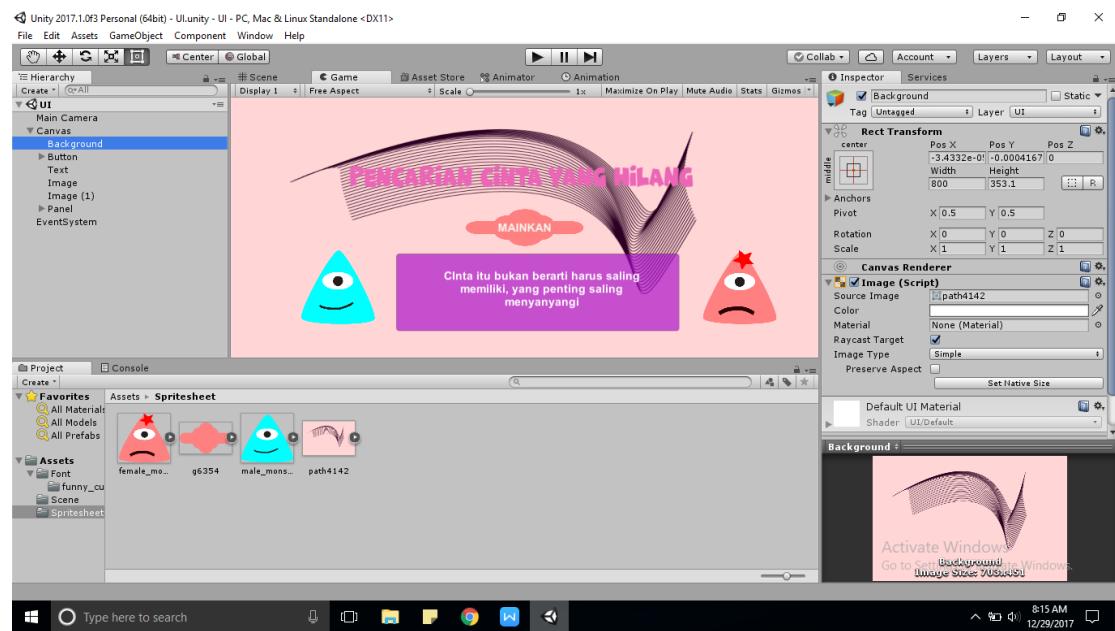
HAND OUT PEMBANGUNAN GAME- PERTEMUAN 3

A. GAMEOBJECT ACTIVE

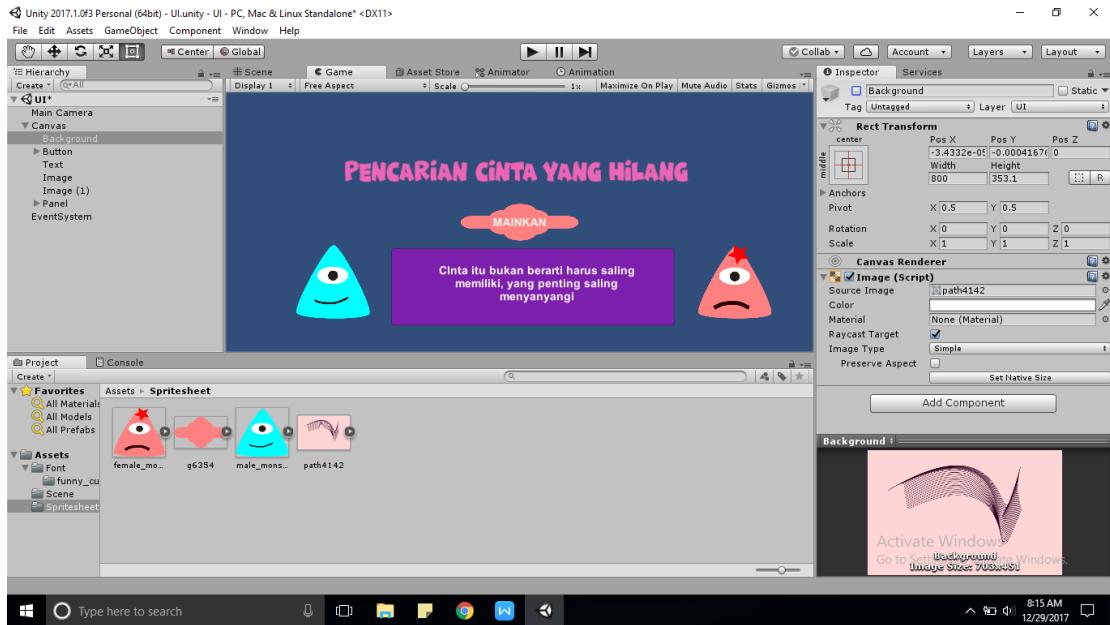
Sebagaimana yang telah dijelaskan sebelumnya bahwa gameobject adalah semua objek yang berada disebuah scene baik yang tampak maupun tidak tampak. Ciri dari sebuah gameobject adalah memiliki Transform. Sebuah gameobject dapat kita atur aktif maupun tidak aktif. Jika kita atur tidak aktif maka walaupun berada di hirearchy namun gameobject tersebut tidak akan muncul maupun dijalankan ketika permainan dimainkan. Untuk mencontohkan kasus ini, buka proyek UI yang kita bangun dibab sebelumnya.



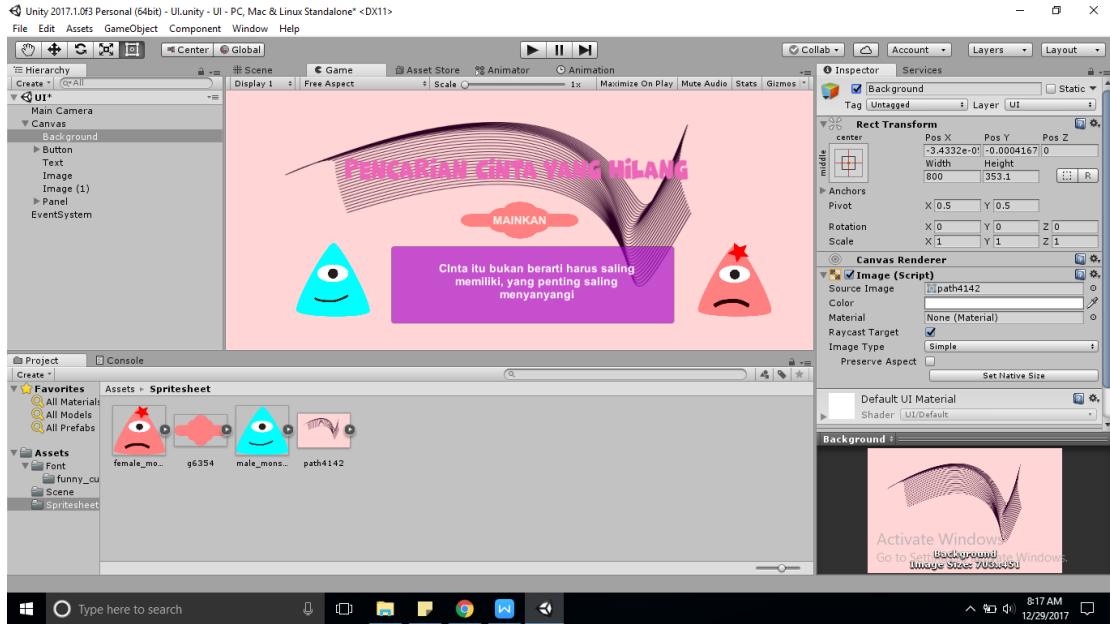
Sekarang pilih gameobject Background



Lalu uncheck setting active/non active gameobject yang berada disebelah nama gameobject

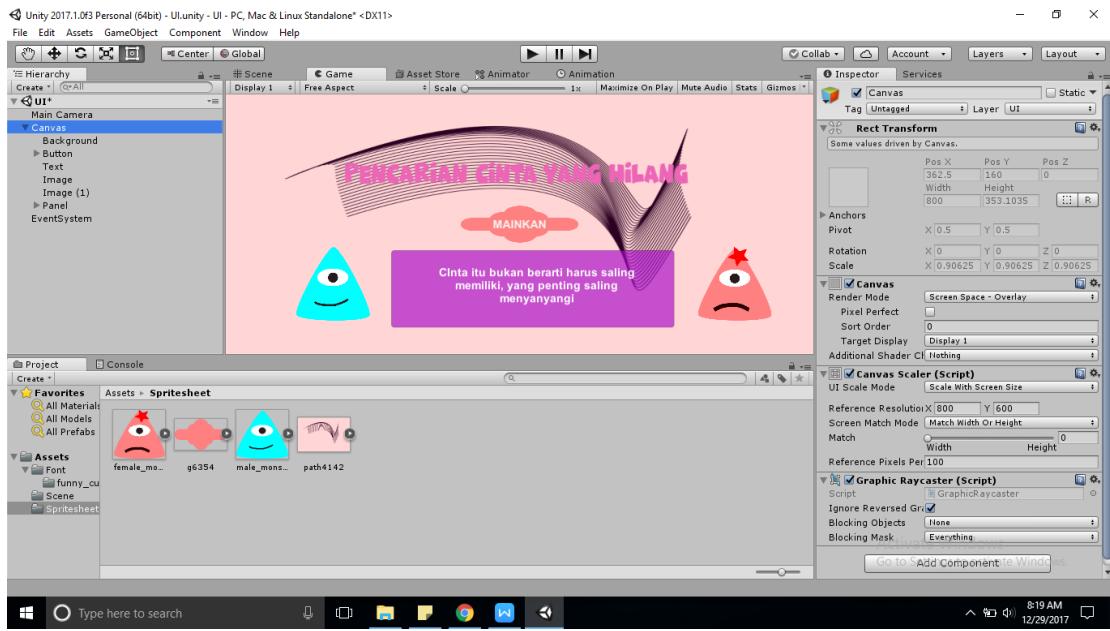


Maka bisa dilihat bahwa background hilang padahal daftar di hierarchy masih ada, namun karena kita setting di hierarchy non active maka menjadi tidak kelihatan. Sekarang coba aktifkan kembali dengan menekan kembali checkbox disebelah nama gameobject.

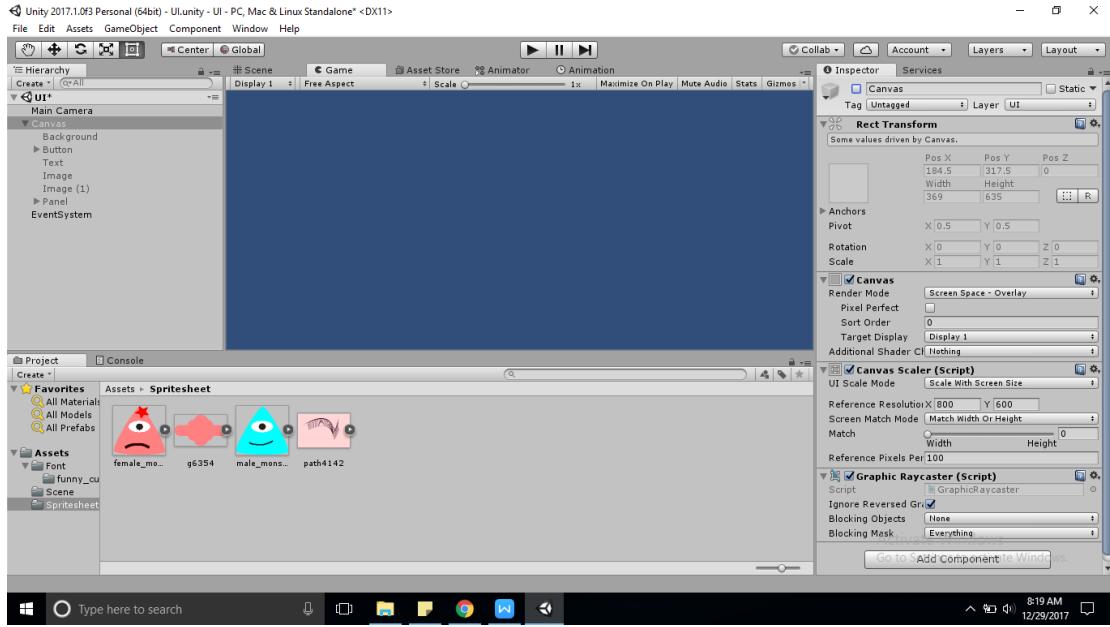


Maka karena gameobject berstatus active maka gameobject pun muncul kembali.

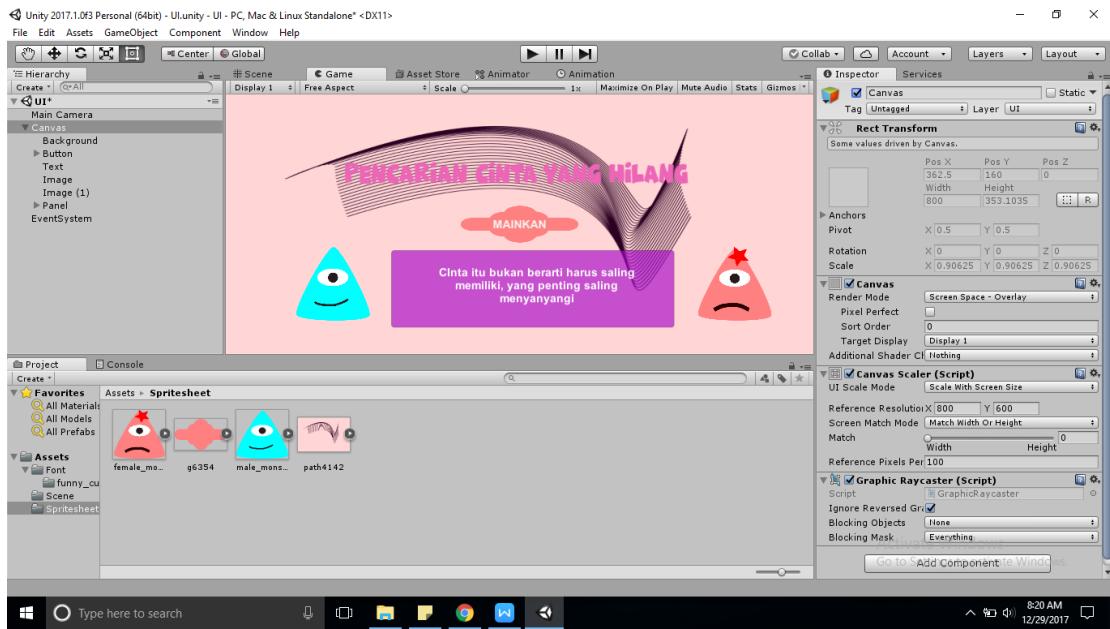
Gameobject parent jika di non activekan maka semua gameobject child dibawahnya akan ikut non active. Untuk mencontohkan ini silahkan pilih canvas.



Lalu atur canvas menjadi non active



Maka semua gameobject child ikut menjadi non active, ditandai dengan tulisan yang awalnya hitam menjadi abu. Jika canvas sebagai parent kita atur active kembali maka



Semua gameobject child akan kembali active ditandai dengan tulisan yang awalnya abu-abu(non active) menjadi hitam (active)

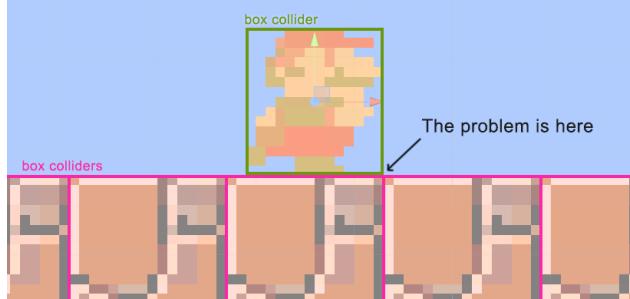
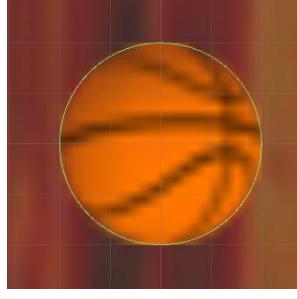
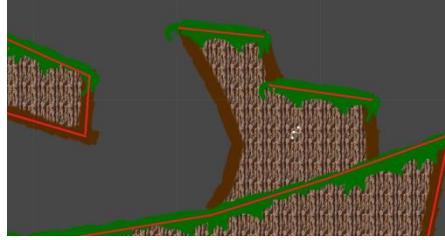
B. GAMEOBJECT COMPONENT

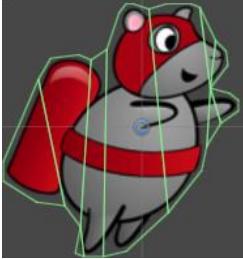
Sebuah gameobject selain memiliki transform juga memiliki kemampuan untuk ditambahkan component. Component adalah fitur-fitur atau kapabilitas yang ditambahkan pada sebuah gameobject. Sebuah gameobject dapat memiliki banyak component, begitupula sebuah component dapat dipasang pada berbagai gameobject. Analogi gameobject dengan component dapat dibayangkan seperti kerangka motor(mewakili gameobject) dengan sparepart (component). Pada awalnya kerangka motor tidak dapat berjalan namun ketika ditambahkan sparepart roda (component) maka motor dapat bergerak dan jika ditambah sparepart lampu (component) maka motor dapat mengeluarkan cahaya... dst

Component yang terdapat di unity sangat banyak, namun berikut ini adalah component-component yang sering digunakan pada pembangunan game 2D

Berikut ini adalah daftar component yang ada di unity

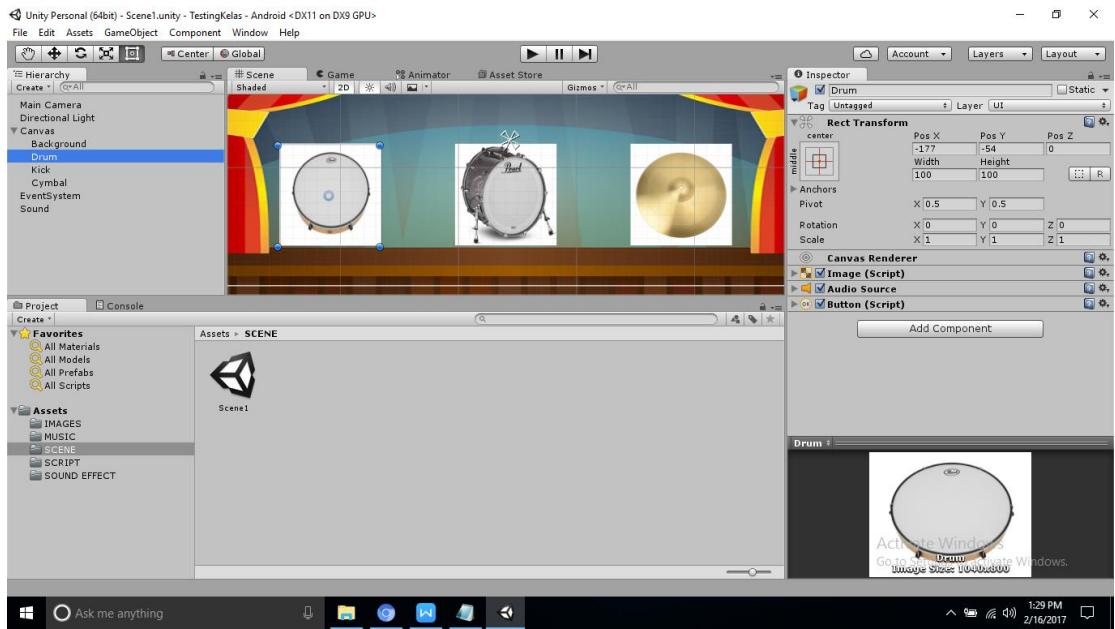
No	Nama Component	Keterangan
1	Script	Komponen yang berisi kode-kode untuk dapat menambahkan fungsi atau perilaku baru terhadap gameobject.
2	RigidBody2d	Rigidbody2d berfungsi untuk menambahkan engine physic pada sebuah sprite sehingga sprite tersebut memiliki berat dan dapat dikoding untuk didorong, dipantulkan, diseret, dsb. Rigidbody2d memerlukan collider untuk mengenali area yang terkena dampak dari

		<p>rigidbody2d tersebut.</p> <p>Secara umum ada 3 body type rigidbody2d yaitu dynamic, static, dan kinematic. yang akan kita pelajari secara bertahap lebih lanjut.</p>
3	Box Collider 2D	<p>Collider adalah component yang mendeteksi terjadinya tumbukan antara sebuah gameobject dengan gameobject lainnya. Box Collider adalah component yang berbentuk kotak untuk mengatur perilaku physic jika terjadi tumbukan</p> 
4	Circle Collider 2D	<p>Sama dengan fungsi collider namun bentuknya bulat, cocok untuk dipasang pada sprite berbentuk bulat seperti bola</p> 
5	Edge Collider 2D	<p>Collider yang digunakan pada sebuah sisi permukaan sebuah sprite saja, cocok digunakan pada sprite permukaan tanah atau bagian2 yang diatur tidak dapat dilewati pemain</p> 
6	Polygon Collider 2D	<p>Collider yang berbentuk bebas dan mengikuti bentuk dari gameobjectnya</p>

		
7	Physics 2D	Component2 yang memberikan fungsi physics (fisika) kepada sebuah gameobject. Beberapa physic yang tergolong 2D antara lain Distance Joint 2D, Fixed Joint 2D, Friction Joint 2D, Hinge Joint 2D, Relative Joint 2D, Slider Joint 2D, Spring Joint 2D, Target Joint 2D, Wheel Joint 2D, Area Effector 2D, Buoyancy Effector 2D, Point Effector 2D, Platform Effector 2D, Surface Effector 2D, dan Constant Force 2D
8	Audio	Component2 yang memberikan fungsi suara kepada gameobject, antara lain Audio Listener dan Audio Source
9	Camera	Component yang membuat sebuah gameobject tampil dilayar
10	Layout	Component2 yang mengatur posisi gameobject di canvas. Yang tergolong disini antara lain Sprite Renderer, Canvas Renderer, GUI Texture, Rect Transform, Canvas, Canvas Group, Canvas Scaler, Layout Element, Content Size Fitter, Aspect Ratio Fitter, Horizontal Layout Group, Vertical Layout Group, dan Grid Layout Group
11	Event	Component2 yang mengatur event. Yang tergolong disini adalah Event System, Event Trigger, dan Physics 2D Raycaster
12	UI	Component2 yang dapat digunakan untuk membangun UI. Yang tergolong disini antara lain Text, Image, Raw Image, Mask, 2D Rect Mask, Button, Input Field, Toggle, Toggle Group, Slider, ScrollBar, Dropdown, Scroll Rect, dan Selectable
13	Component Lainnya	Dan masih banyak component2 lain yang bisa digunakan untuk membangun game

C. COMPONENT STACK

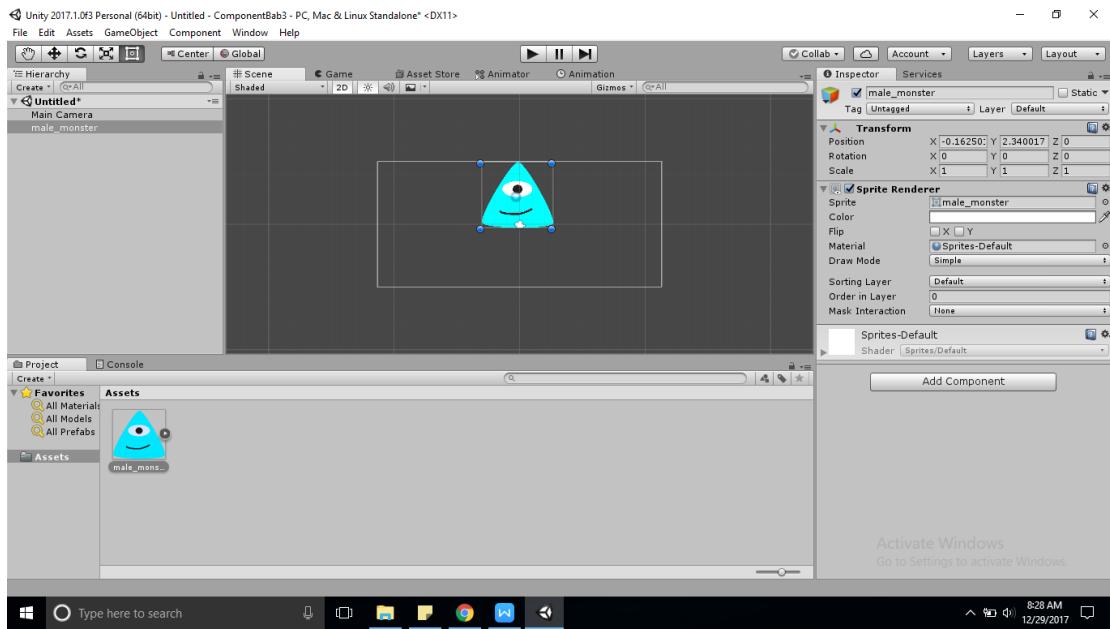
Sebuah gameobject dapat memiliki beberapa component untuk mencapai fungsionalitas yang diinginkan, sejumlah component yang dipasang pada sebuah gameobject disebut sebagai component stack. Misalkan kita ingin membuat sebuah gambar drum yang jika ditekan mengeluarkan suara drum, maka berikut ini adalah component2 yang dibutuhkan.



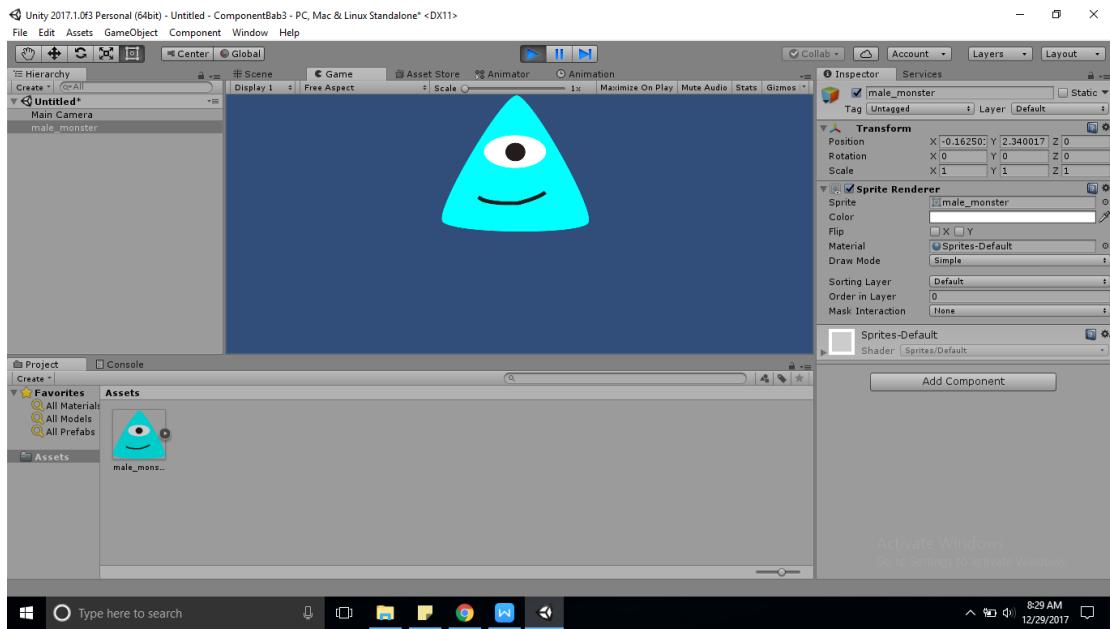
Dapat dilihat diatas bahwa component yang dipasang adalah:

1. Rect Transform, component untuk memposisikan gambar di canvas
2. Canvas Renderer, component untuk menampilkan gambar di canvas
3. Image, component untuk menyimpan gambar ke gameobject
4. Audio, component untuk mengeluarkan suara
5. Button, component untuk gameobject dapat diklik

Untuk mencoba mempraktekan hal ini, maka buatlah sebuah project/scene baru, sekarang kita akan menggunakan area default/area main camera. Disiapkan pula satu gambar dan tambahkan ke scene, misalkan disini ditambahkan gambar monster imut yang dibuat pada bab lalu. Tempatkan diposisi atas tapi jangan melewati area garis area main camera



Ketika dicoba play dengan menekan tombol diatas maka tidak terjadi apapun.



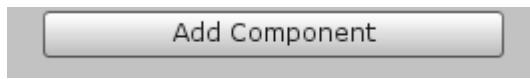
Kita akan coba menambahkan component sebelumnya jangan lupa untuk keluar dulu dari mode permainan karena jika kita melakukan perubahan apapun ketika mode permainan maka semua perubahan tersebut tidak akan tersimpan. Mode permainan ditandai dengan tombol permainan yang biru seperti berikut:



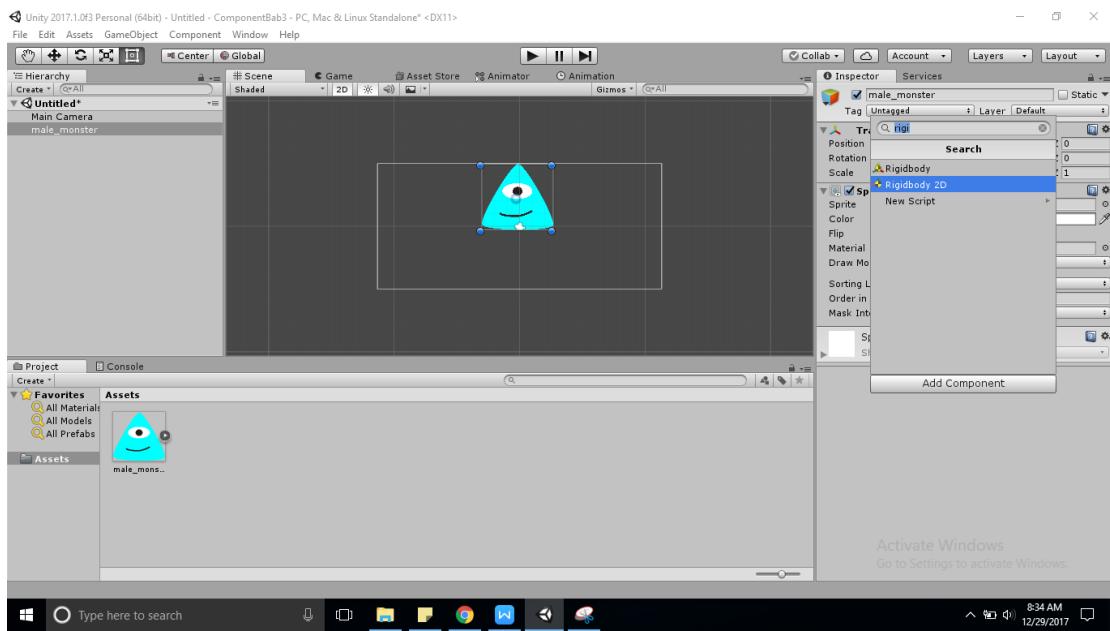
Sedangkan mode edit tombolnya berwarna hitam sebagai berikut



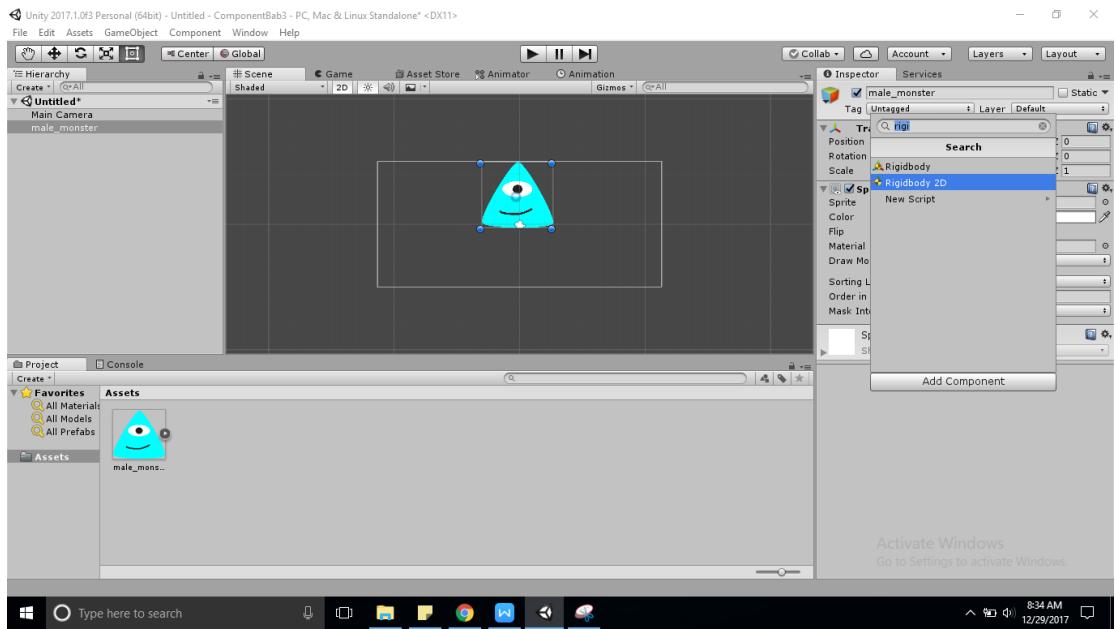
Nah sekarang kita coba tambahkan component rigidbody 2d yang mampu memberikan fungsi physic kepada gameobject yang dipasang component tersebut. Pilih gameobject yang ingin ditambahkan component, tekan tombol add component



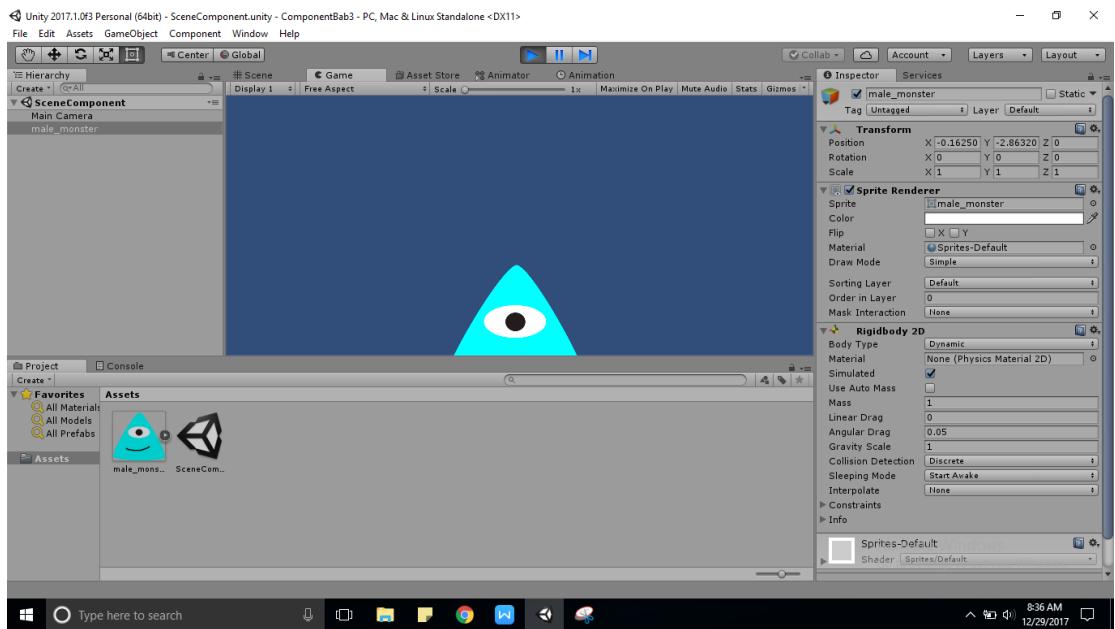
Lalu cari component yang ingin ditambahkan, pada kasus ini adalah Rigidbody2D



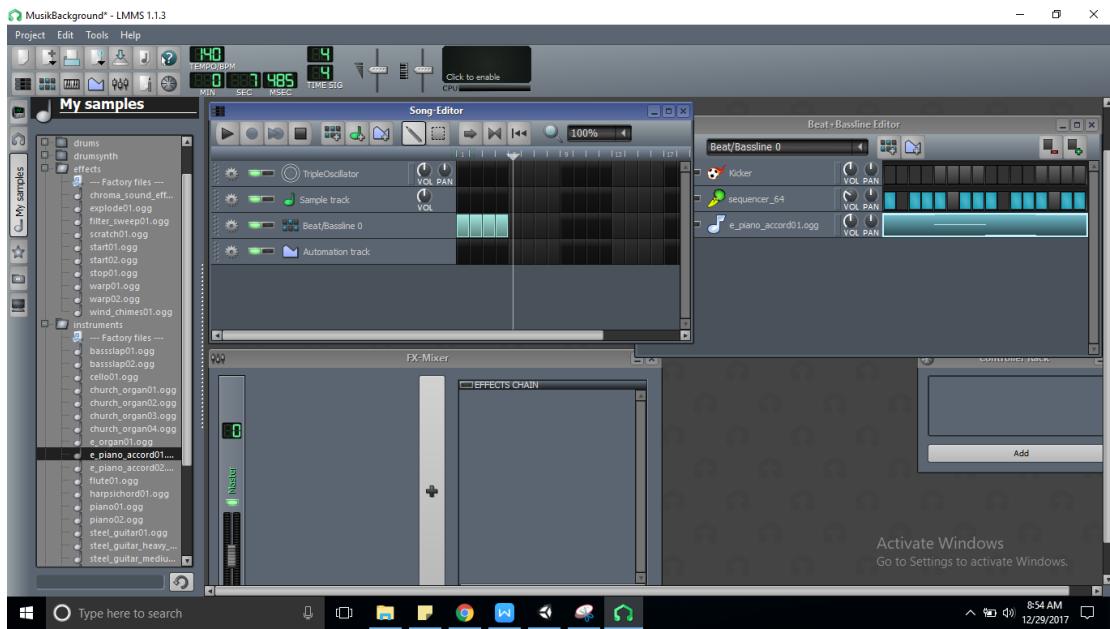
Hasilnya adalah seperti berikut:



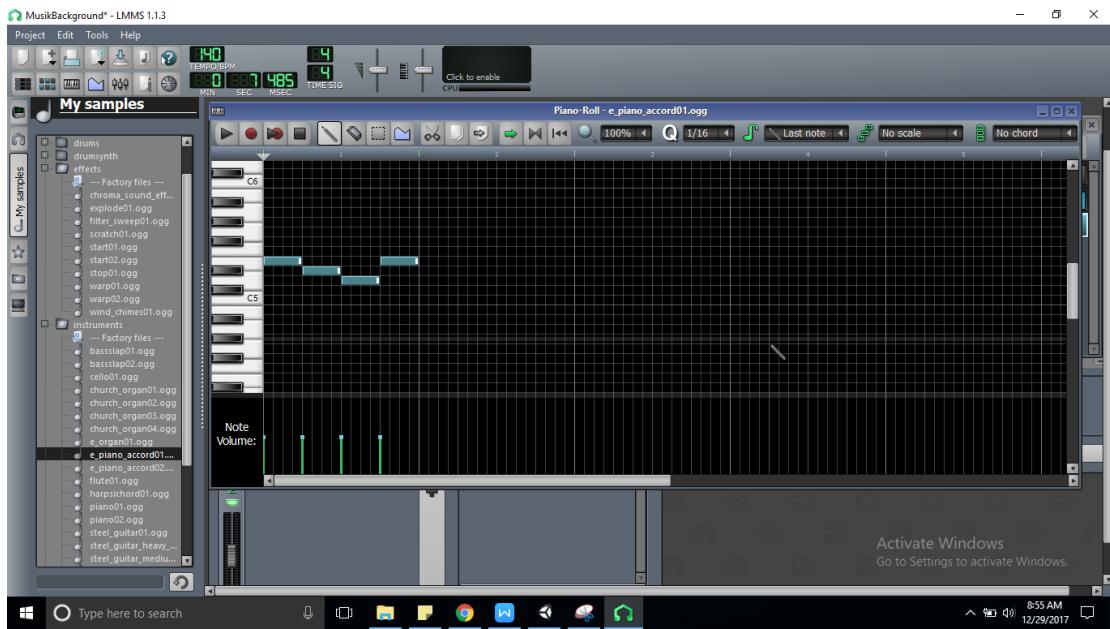
Sekarang jika kita mainkan maka gambar kita akan turun kebawah, disebabkan penambahan component rigidbody menyebabkan gameobject gambar memiliki berat yang dapat ditarik gravitasi



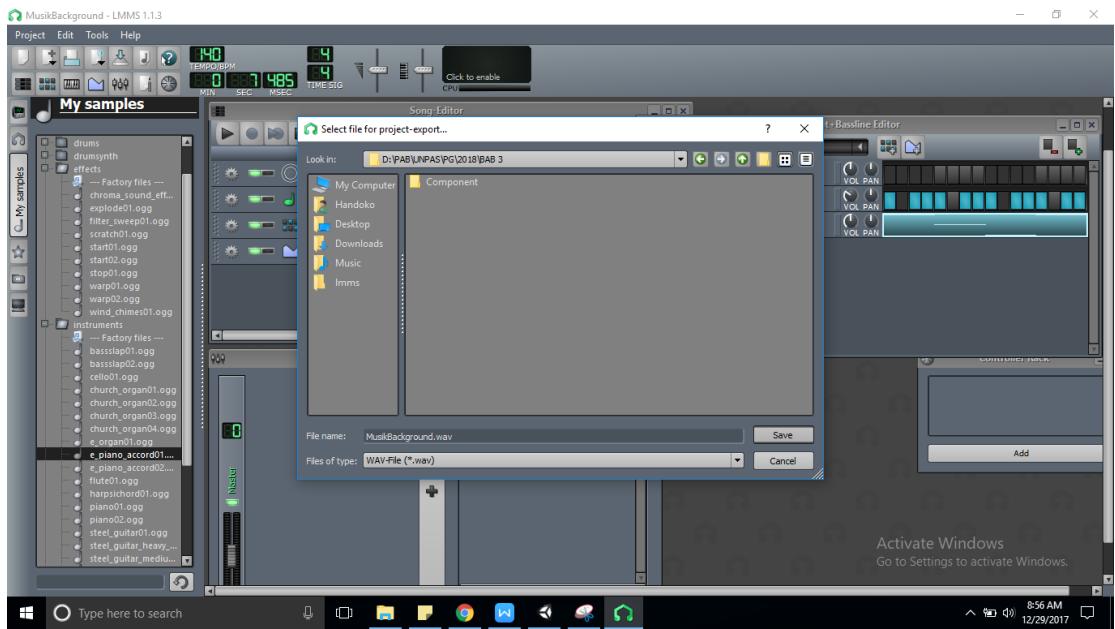
Kita bisa menambahkan berbagai macam gameobject dimana masing2 gameobject tersebut kita tambahkan component yang berbeda masing2 sesuai peruntukannya. Misalkan kali ini gambar kita akan turun kebawah sambil diiringi musik. Sebelumnya siapkan musik baik dari download maupun buat sendiri, pada bab ini dicoba membuat musik sendiri menggunakan Imms sebagai berikut, perhatikan Song Editor dan juga Bassline Editor



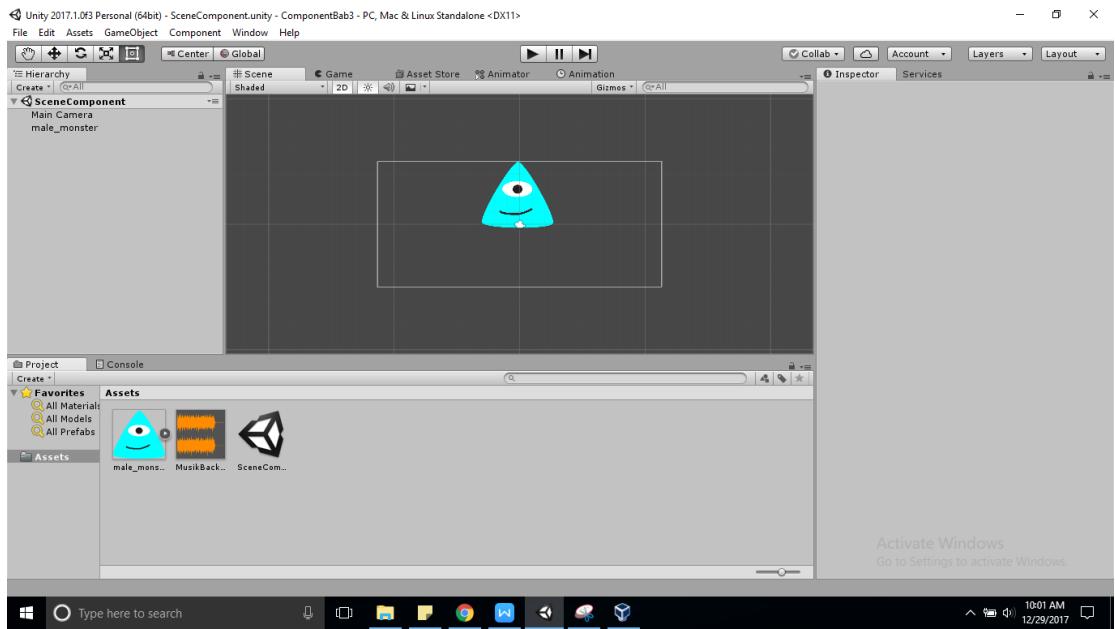
Berikut adalah konfigurasi pada piano accord



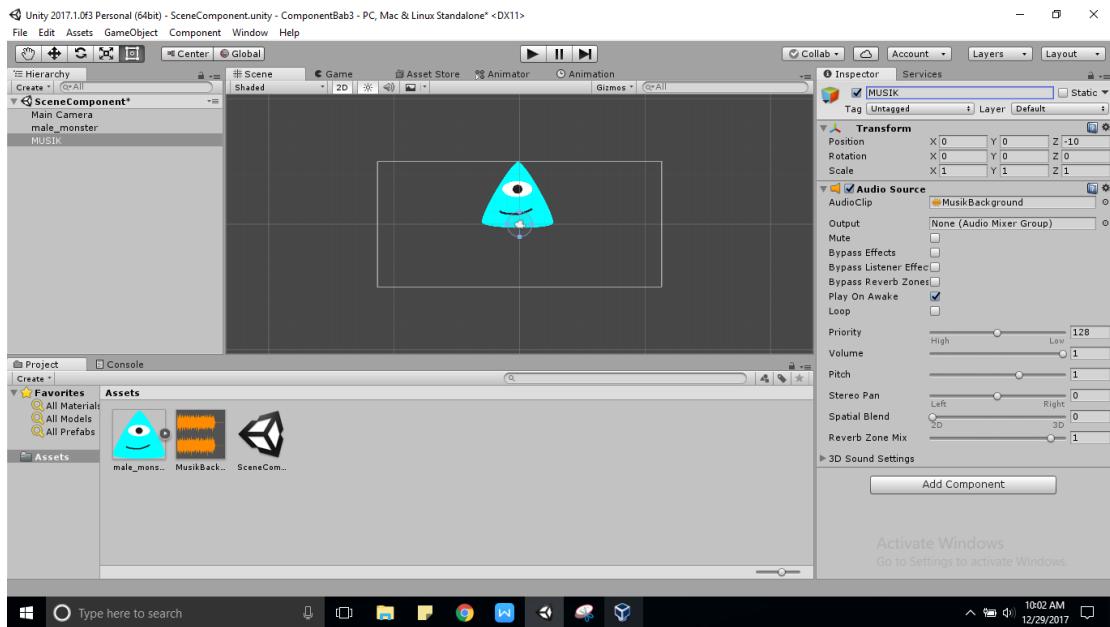
Anda bisa membuat musik sendiri sesuai selera anda, diatas hanyalah contoh, setelah semua dirasa siap maka kita export



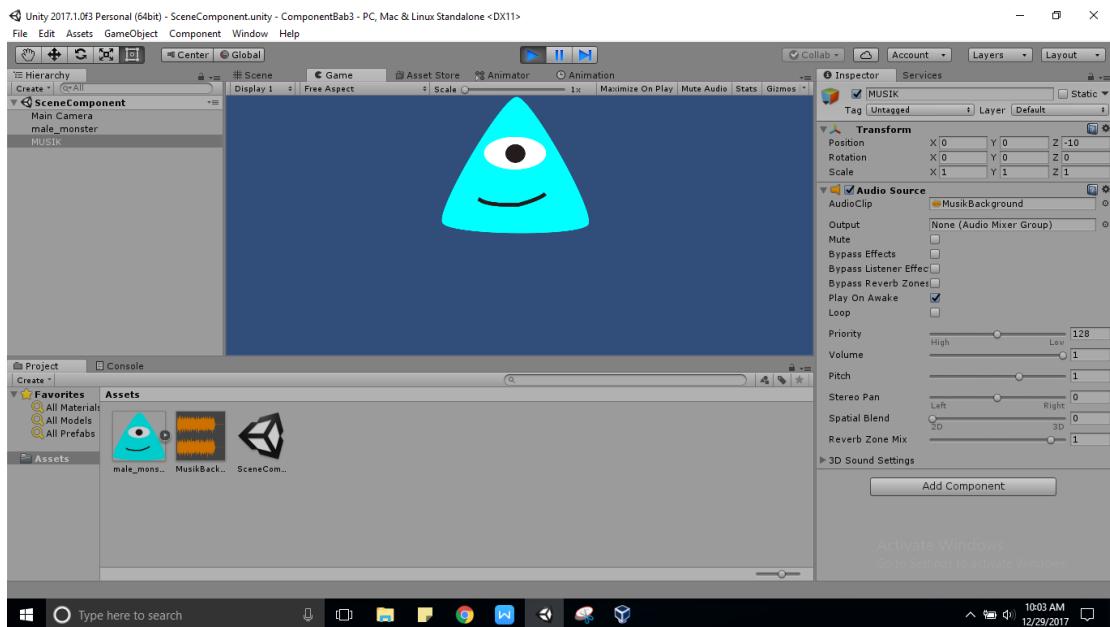
Tempatkan musik yang baru anda buat di project asset



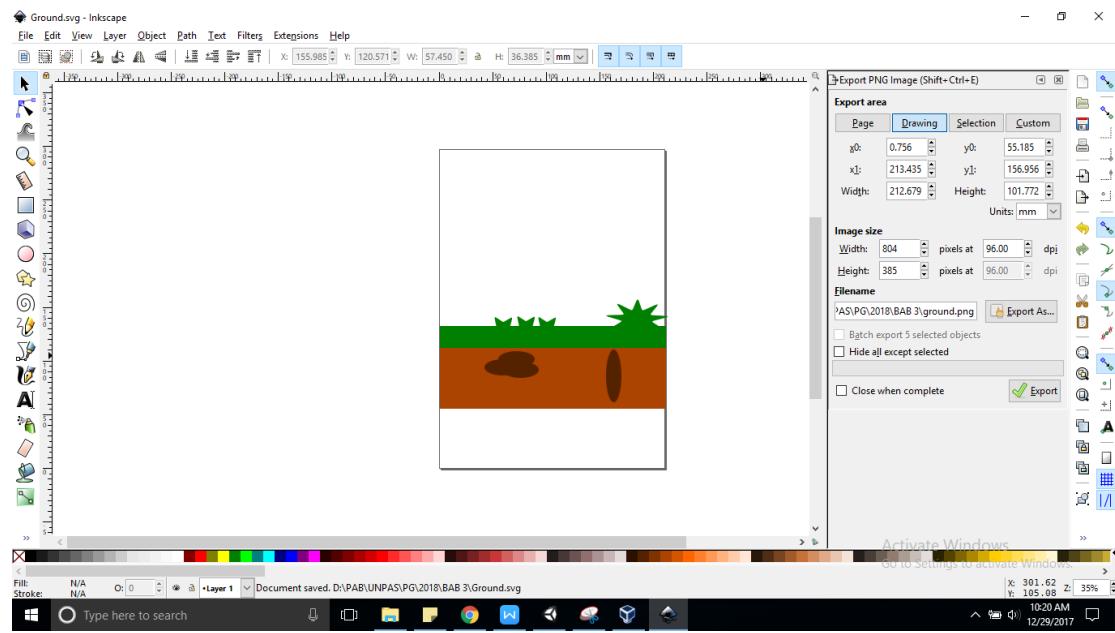
Tambahkan gameobject baru, beri nama musik lalu add component audio source. Drag And Drop musik yang baru ditambahkan ke isian Audio Clip pada component Audi Source



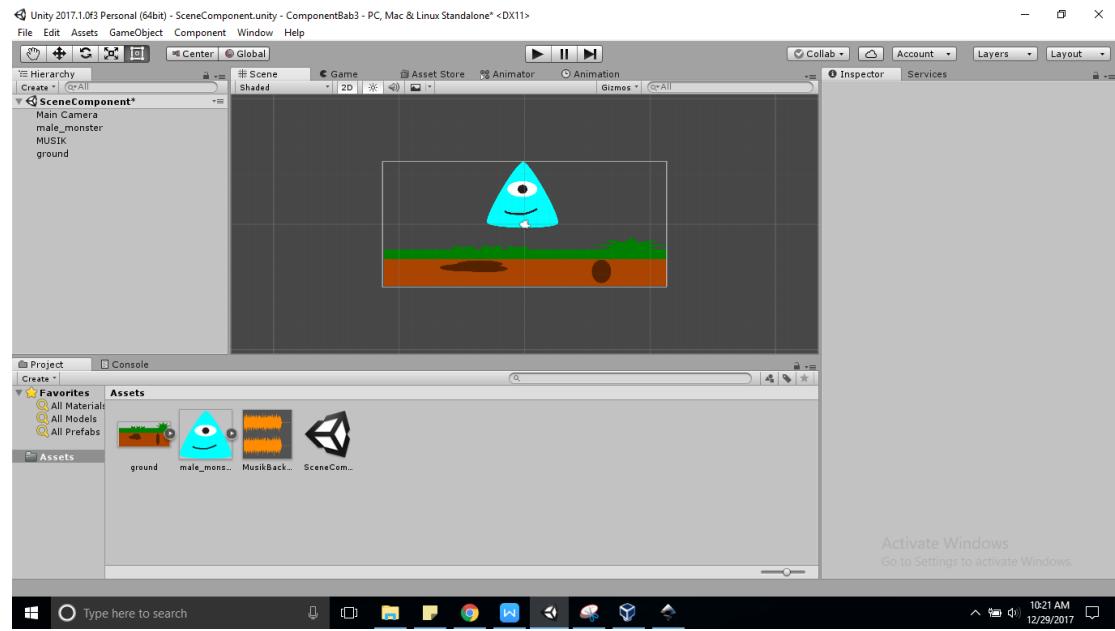
Silahkan dimainkan, bisa dilihat bahwa sekarang game menjalankan karakter yang jatuh sambil diiringi musik



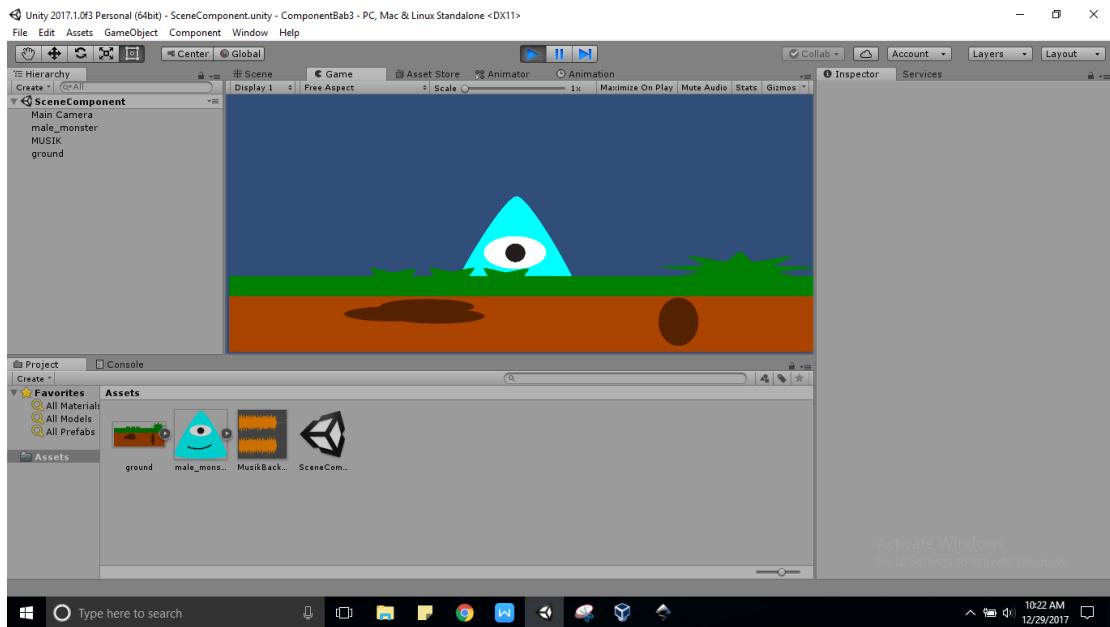
Selanjutnya component pada sebuah gameobject dapat mempengaruhi perilaku gameobject yang lain. Misalkan monster yang jatuh terus kebawah dapat kita halangi dengan menambahkan gambar tanah dibawahnya sehingga monster ketika menginjak tanah maka akan diam diatas tanah dan tidak akan turun terus dikarenakan tanah ini telah kita tambahkan dan atur component2 yang mampu menghalangi gambar monster turun kebawah. Agar memahaminya, mari kita praktikan. Misalkan kita buatkan gambar tanah di inkscape.



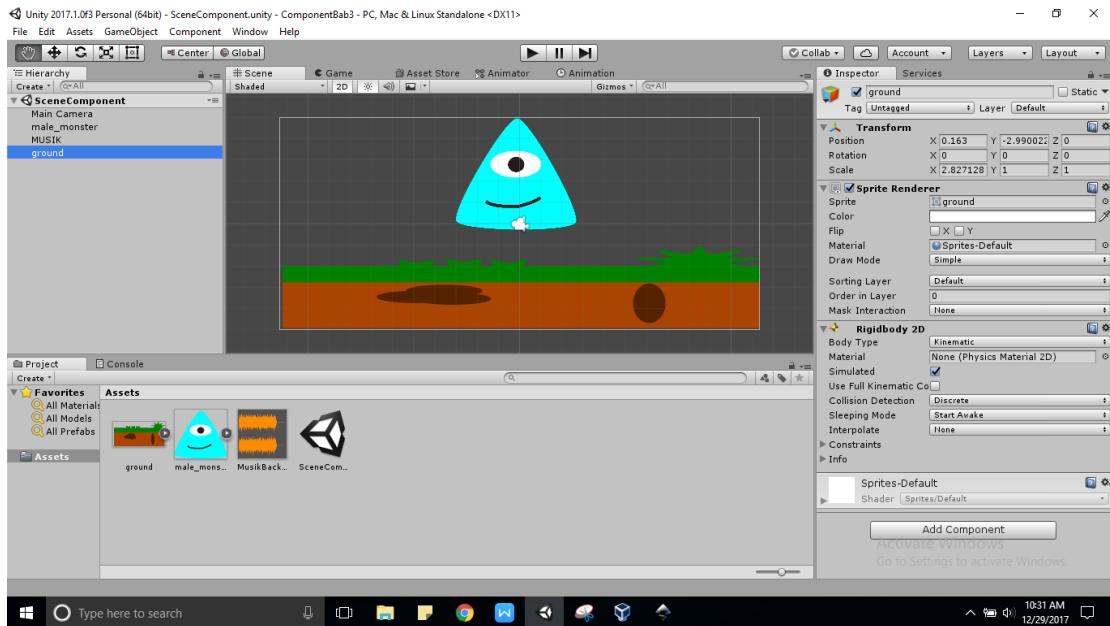
Tambahkan gambar tersebut ke project asset unity, atur di scene seperti berikut



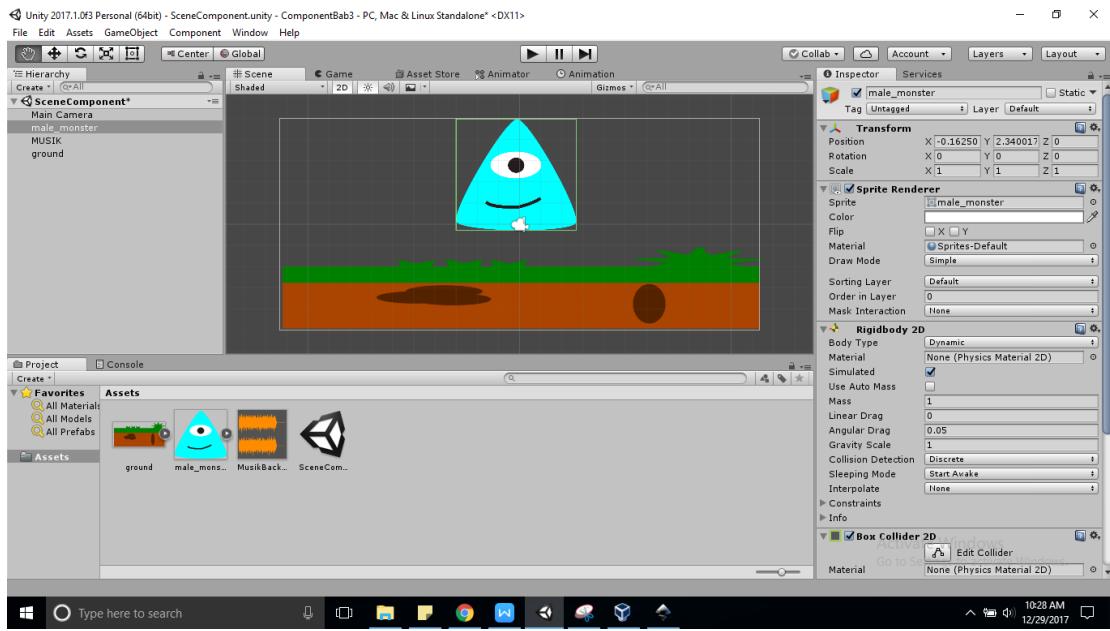
Sekarang walaupun kita jalankan maka gambar monster akan tetap menembus kebawah



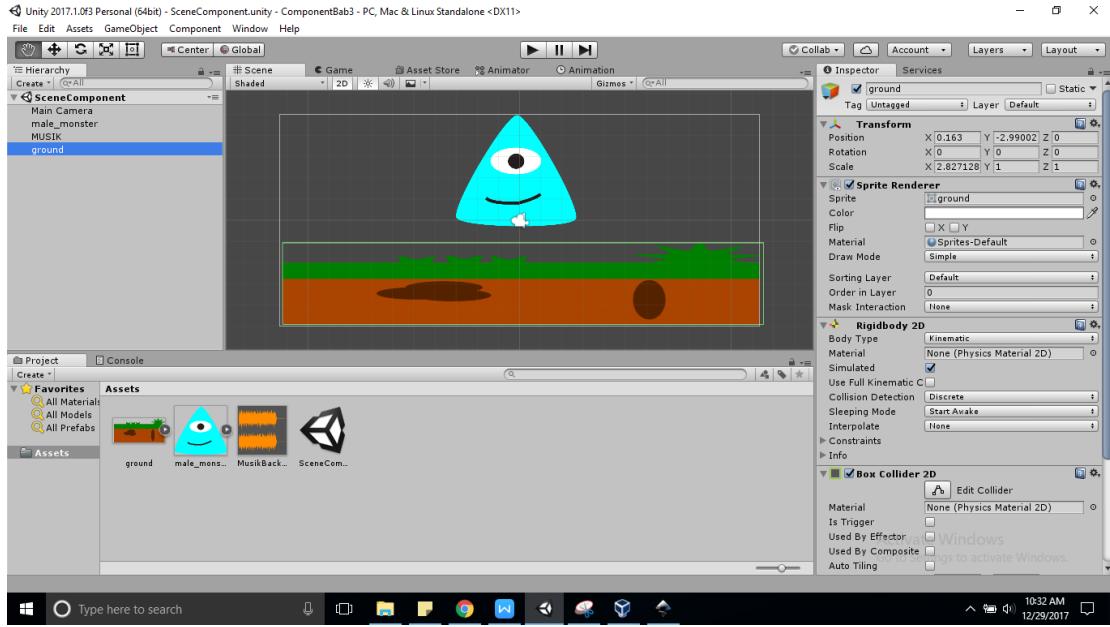
Hal ini dikarenakan kita belum memberikan component rigidbody2d kepada tanah, component rigidbody2d memberikan efek physic sehingga ketika gambar monster mengenai tanah maka dikenali sebagai objek nyata yang tidak bisa ditembus, hal ini berlaku untuk keperluan lain seperti tembok misalkan. Satu hal yang perlu diperhatikan adalah tanah tidak boleh turun ditarik gravitasi ketika ditambahkan rigidbody2d, untuk menghindari ini maka kita rubah body type tanah/ground menjadi kinematic.



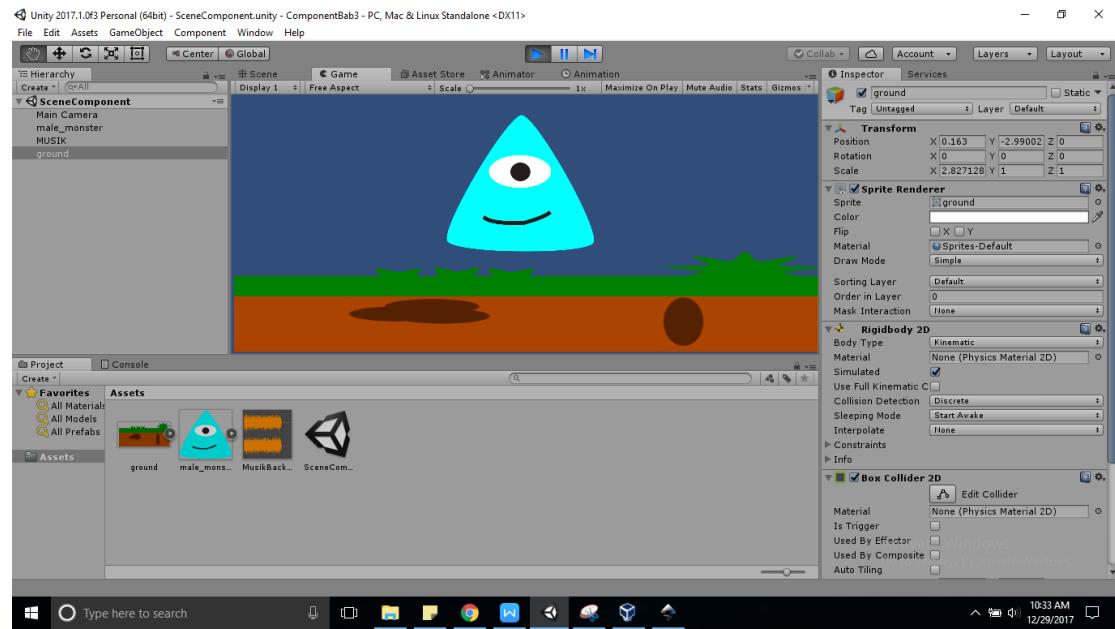
Coba mainkan maka akan didapat hasil gambar monster masih turun kebawah, mengapa? Hal ini dikarenakan rigidbody2d belum mendeteksi sebesar apa area yang memiliki dampak physic? Area penanda dampak physic tersebut dikenal sebagai boxcollider2d, baik monster maupun tanah harus memiliki untuk memberitahu game engine masing2 area physic dari gameobject. Pertama kita tambahkan untuk monster untuk boxcollider2d perhatikan muncul area hijau di sekitar monster itulah boxcollider2D.



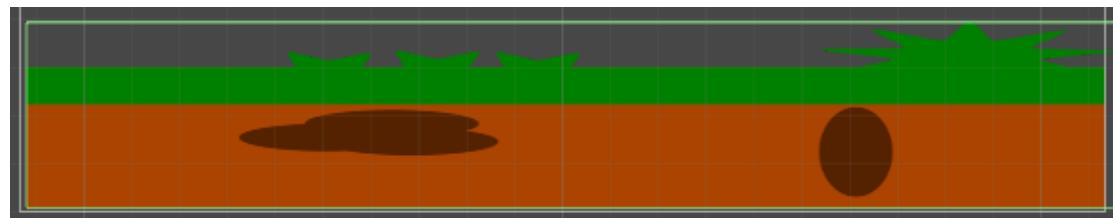
Lakukan hal yang sama untuk tanah



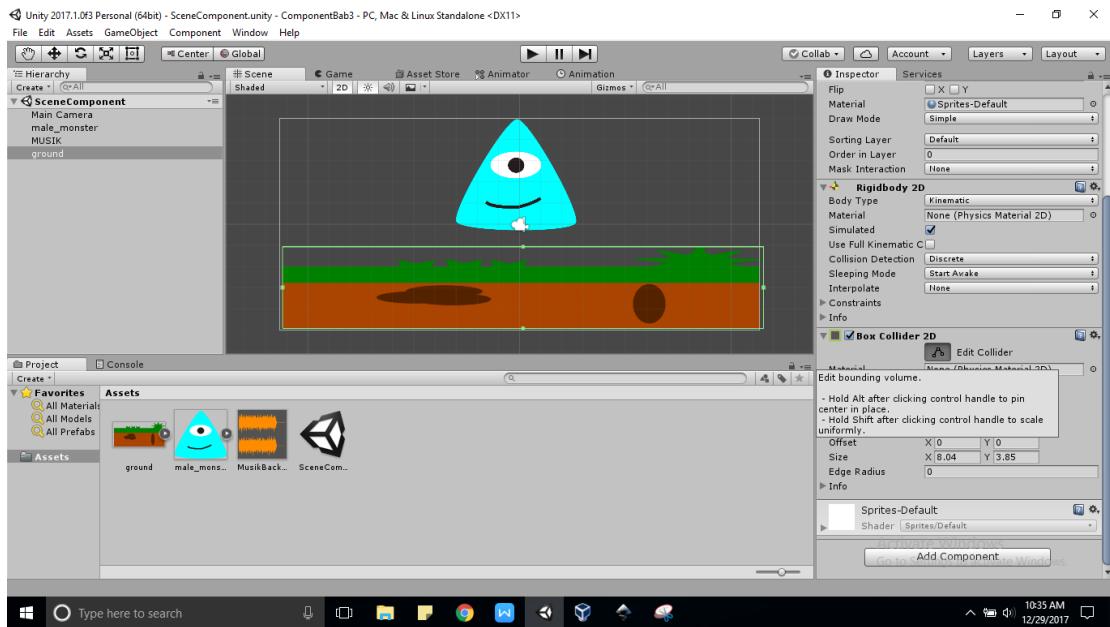
Silahkan dimainkan, maka monster tidak akan jatuh lagi, ketika mengenai tanah



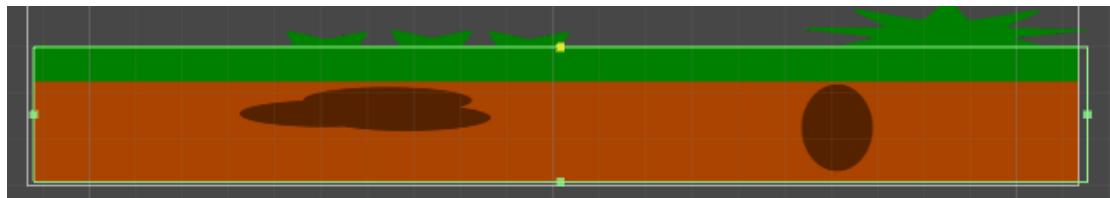
Namun masalahnya karakter monster kita terlihat mengambang, hal ini dikarenakan boxcollider2d pada tanah terlalu tinggi, maka rubahlah collider tanah yang awalnya seperti ini (perhatikan area hijau):



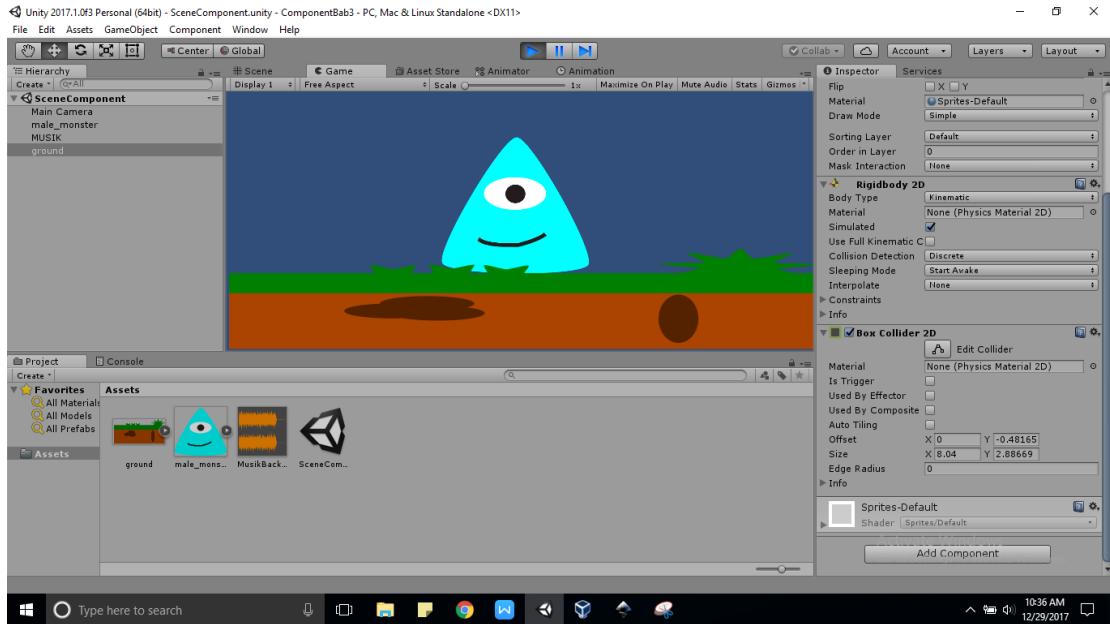
Pilih ground lalu tekan edit collider



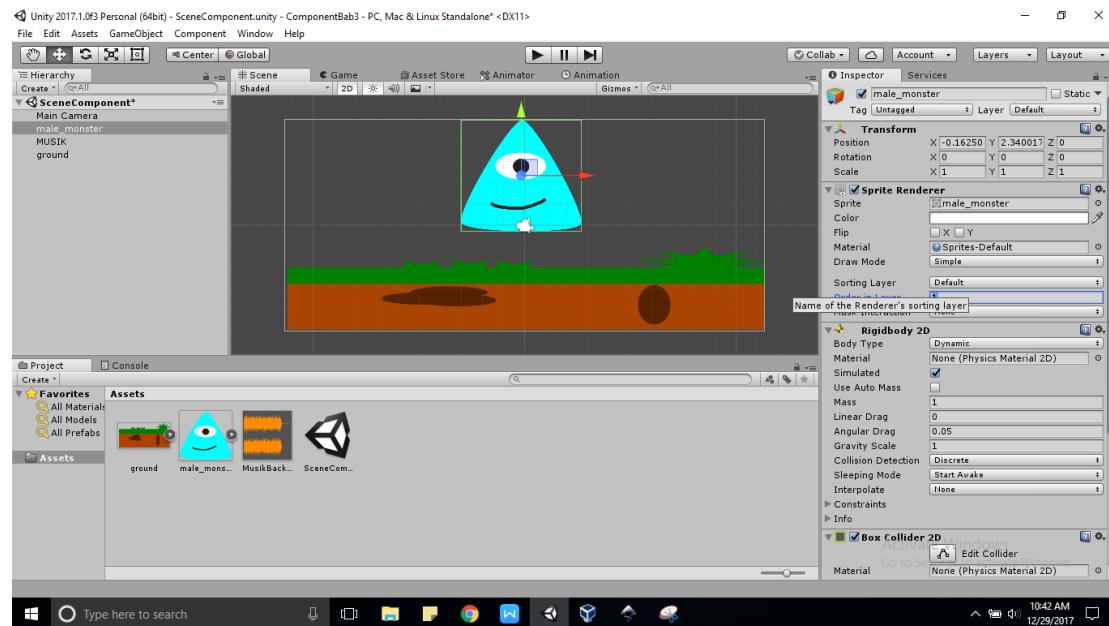
Silahkan sesuaikan, misalkan menjadi seperti ini



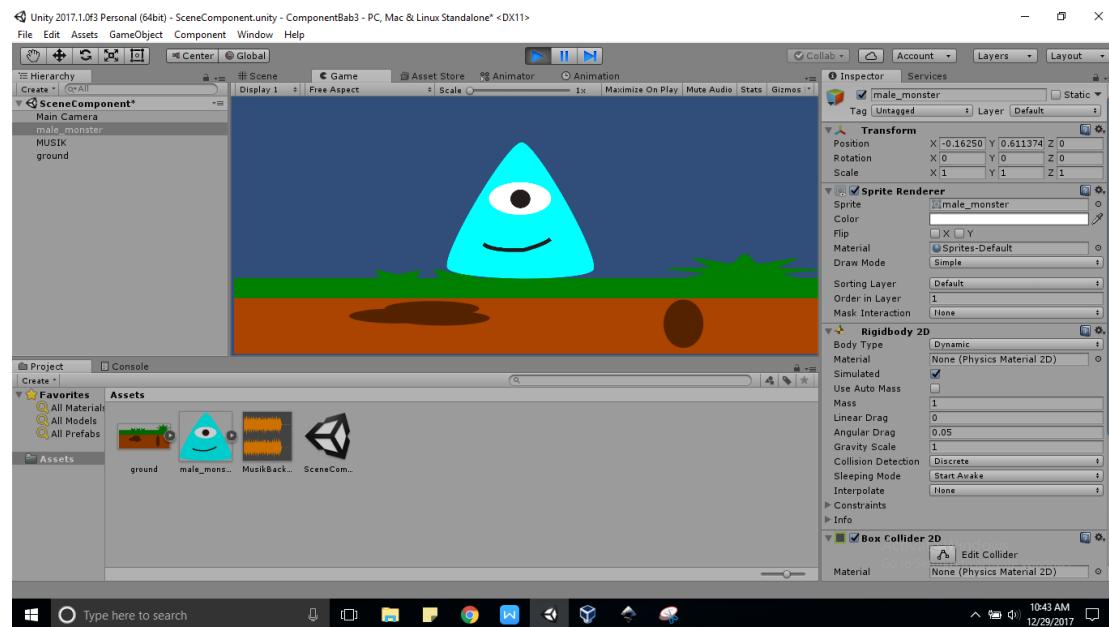
Mainkan dan lihat hasilnya



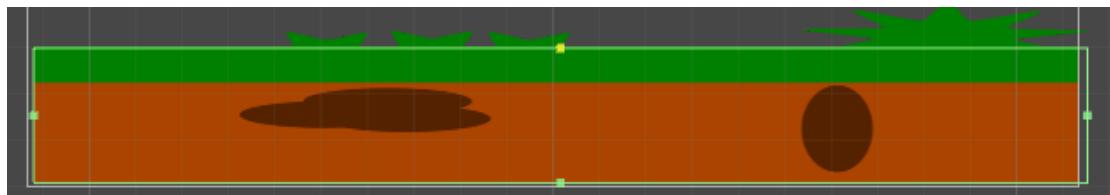
Efek kombinasi antara monster dan tanah diatas menimbulkan pesan bahwa rumput lebat dan monster akan berjalan melewatinya. Jika ingin monster berada didepan rerumputan maka pada component sprite renderer kita ubah isian Order In Layernya dimana objek yang berada didepan harus memiliki nilai lebih besar dibandingkan objek dibelakangnya. Misalkan ground(tanah) 0 karena ini sudah default maka tidak akan kita rubah apapun sedangkan monster kita atur menjadi 1 Order In Layernya, seperti ini:



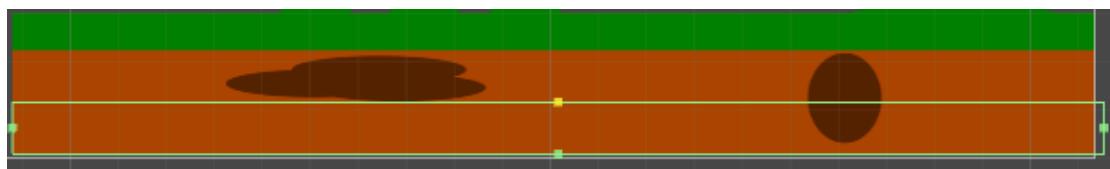
Maka hasilnya adalah sebagai berikut



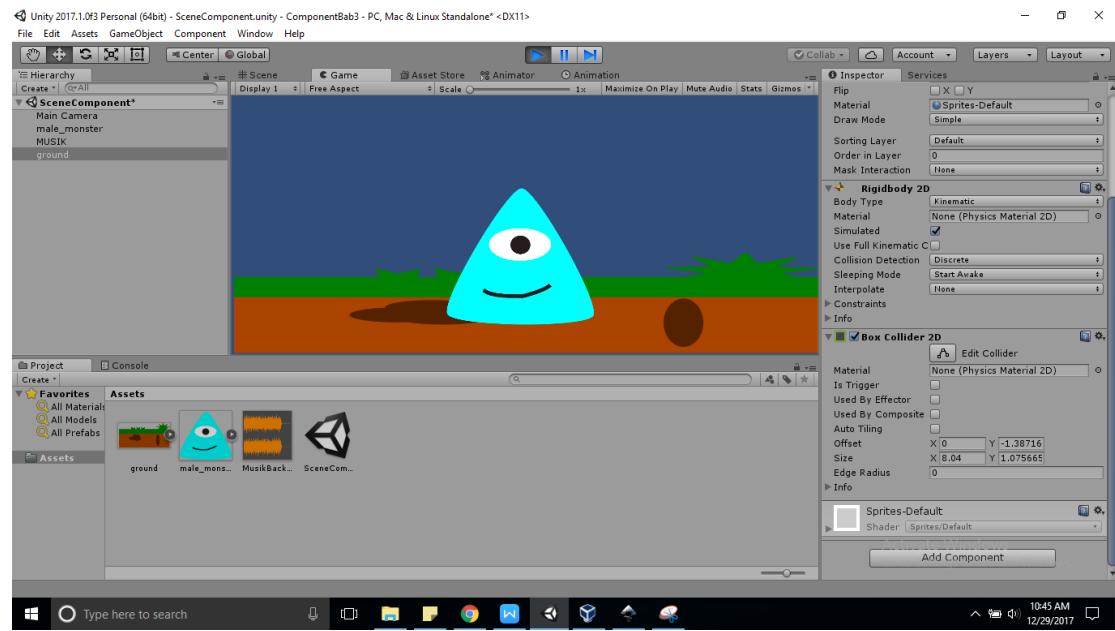
Bisa terlihat bahwa pesannya rumput itu kecil dan pendek sehingga terinjak oleh monster. Kita juga dapat memberi pesan bahwa rumput itu ada diseberang monster, caranya edit collider pada ground kembali yang awalnya seperti ini:



Menjadi seperti ini:



Maka ketika kita mainkan hasilnya sebagai berikut:



Jadi bisa disimpulkan, pada unity gambar hanyalah sesuatu yang abstrak yang menentukan seperti apa gambar tersebut difungsikan bukanlah pada gambarnya melainkan pada component dan pengaturannya. Demikian semoga paham.

Sekarang untuk component stack, artinya sebuah gameobject dapat memiliki lebih dari satu fungsi sesuai dengan component yang ditambahkan. Misalkan gambar monster yang jatuh juga dapat kita hilangkan jika kita menambahkan component dan mengaturnya. Jadi selain memiliki component rigidbody2d menyebabkan memiliki massa sehingga turun kebawah, punya component boxcollider untuk memberitahu area physic, kita tambahkan lagi component baru dan seterusnya, inilah yang dimanakan component stack dimana component demi component bisa kita tambahkan yang secara praktis hampir tidak terbatas dimana component-component tersebut menambahkan kemampuan kepada gameobject tersebut.

Prefab

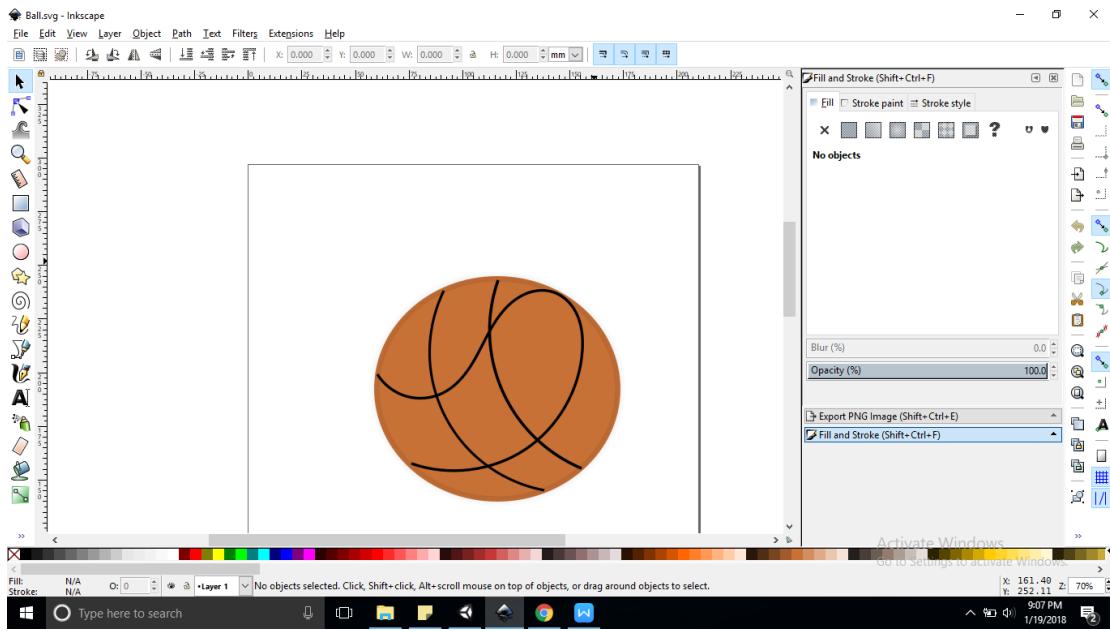
Hal menarik yang akan kita pelajari berikutnya adalah prefab. Prefab adalah gameobject yang disimpan lengkap dengan component dan propertinya. Prefab sangat berguna untuk menampilkan gameobject yang berperilaku sama jika lebih dari satu.



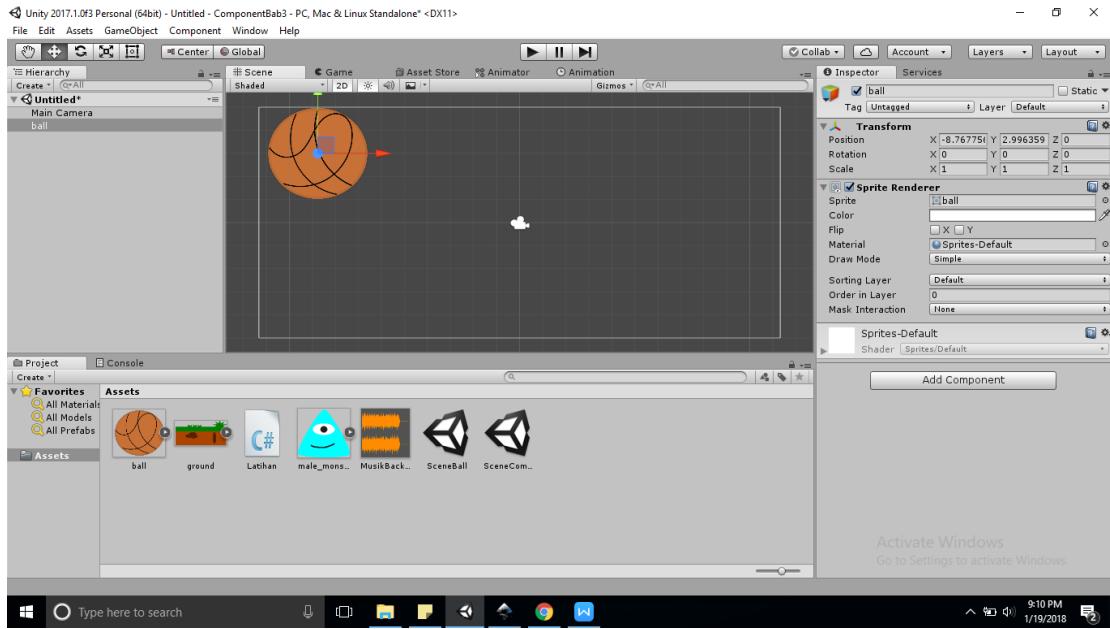
Contoh prefab adalah koin-koin diatas. Agar bisa memiliki fungsi seperti coin yang jika kita tangkap hilang dan menambah jumlah koin maka tentu saja memerlukan komponen yang lebih dari satu, belum lagi kode program yang harus dipasang pada masing-masing koin tersebut untuk mengenali bahwa koin itu tersentuh maka tentu saja repot jika kita pasang masing-masing gameobject di scene lalu kita tambahkan koding dan komponen satu persatu belum lagi mengatur propertinya. Disinilah manfaat daripada prefab.

Untuk dapat memahami penggunaan prefab, mari kita mempraktekannya dengan tahapan sebagai berikut:

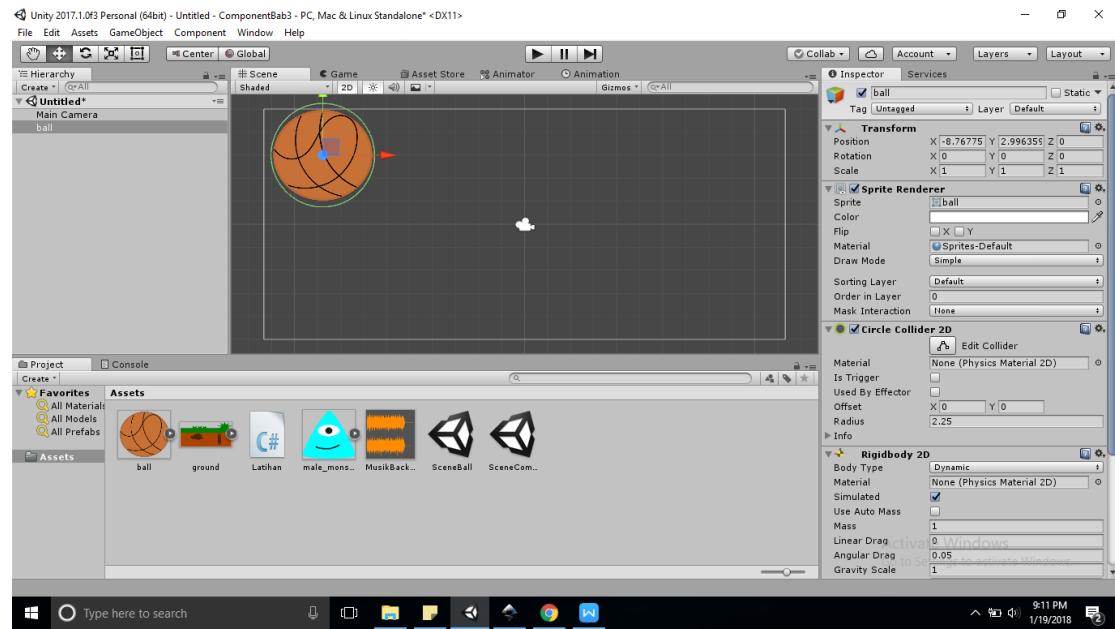
Pertama buka inkscape dan buat bola, pada contoh dibuku ini bolanya adalah bola basket



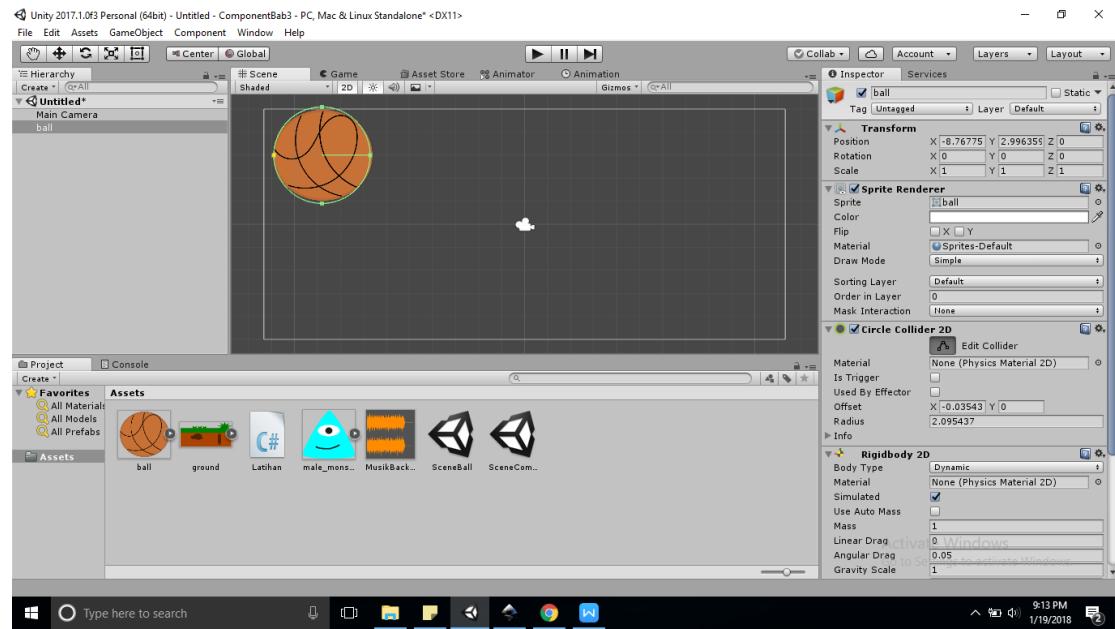
Export menjadi png, dimana backgroundnya harus transparan. Import bolanya ke unity. Buat scene baru dan tempatkan asset bola menjadi sebuah gameobject di scene baru tersebut. Hasilnya adalah sebagai berikut



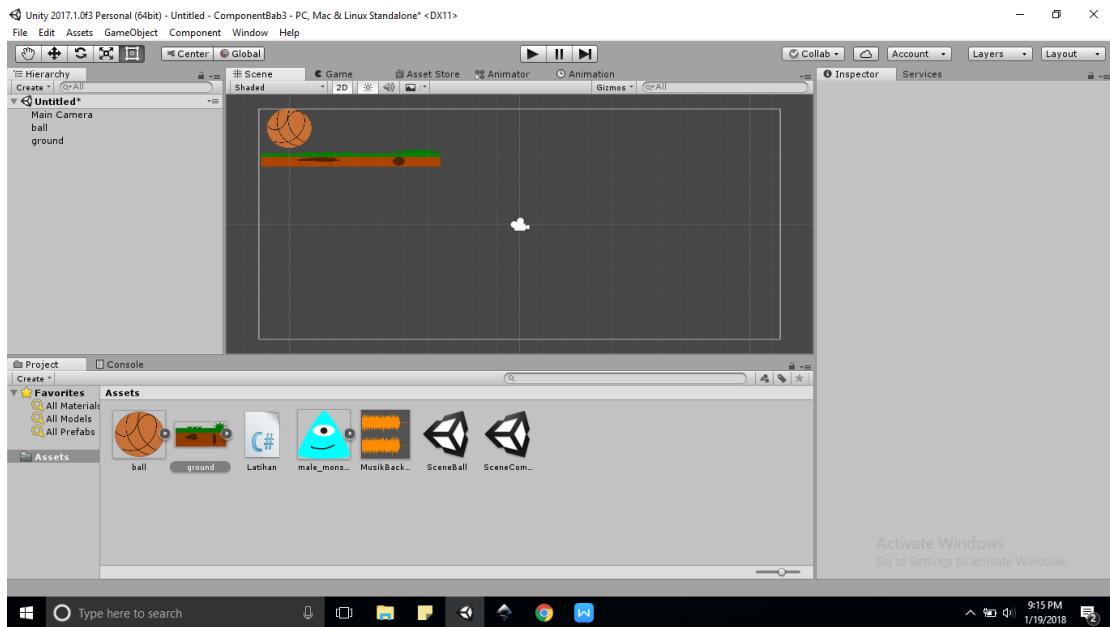
Sekarang skenarionya kita ingin bola diatas bisa jatuh dan menggelinding, maka kita tambahkan rigidbody2d dynamic dan circle collider 2d. Hasilnya seperti ini:



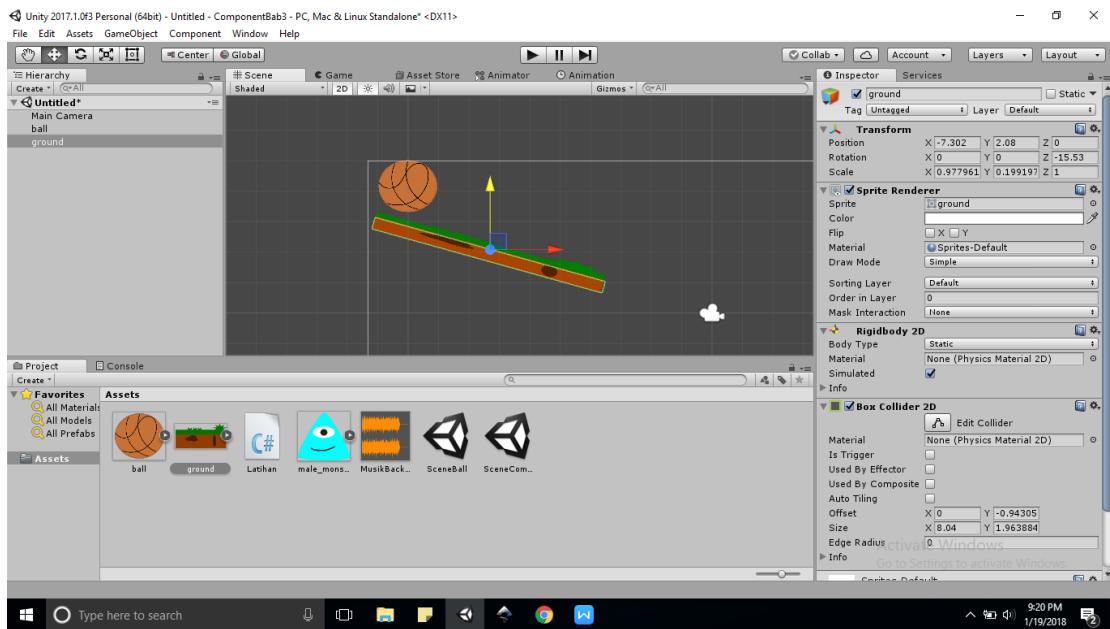
Sesuaikan area collidernya dengan merubah radius dan jika perlu edit collider:



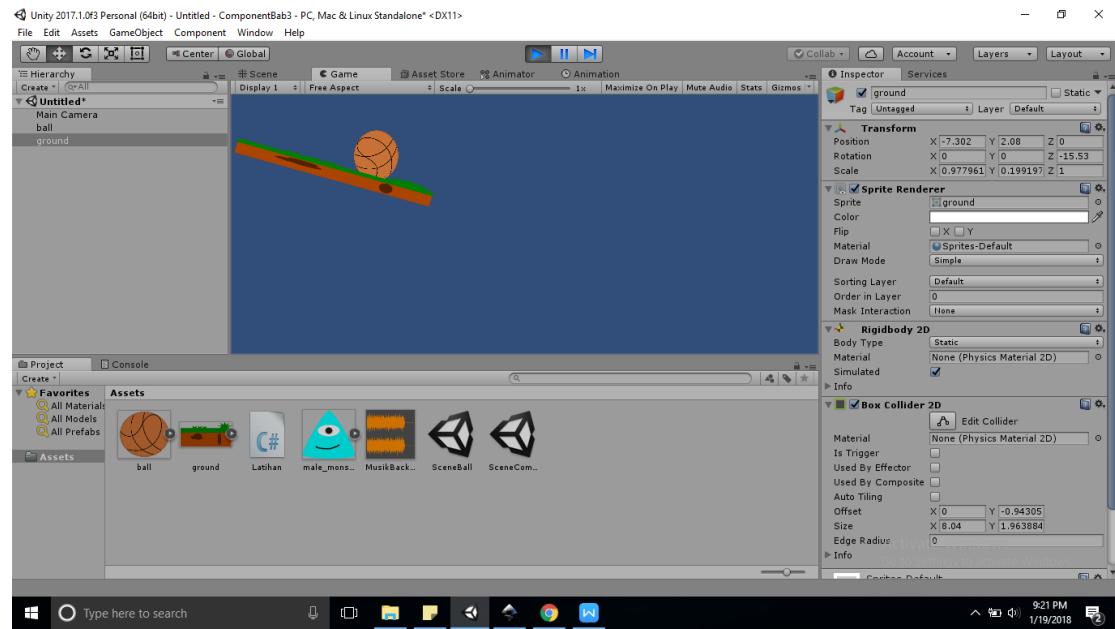
Jika sudah selanjutnya tambahkan ground, dan perkecil ukurannya sehingga perbandingan dengan area main camera adalah sebagai berikut:



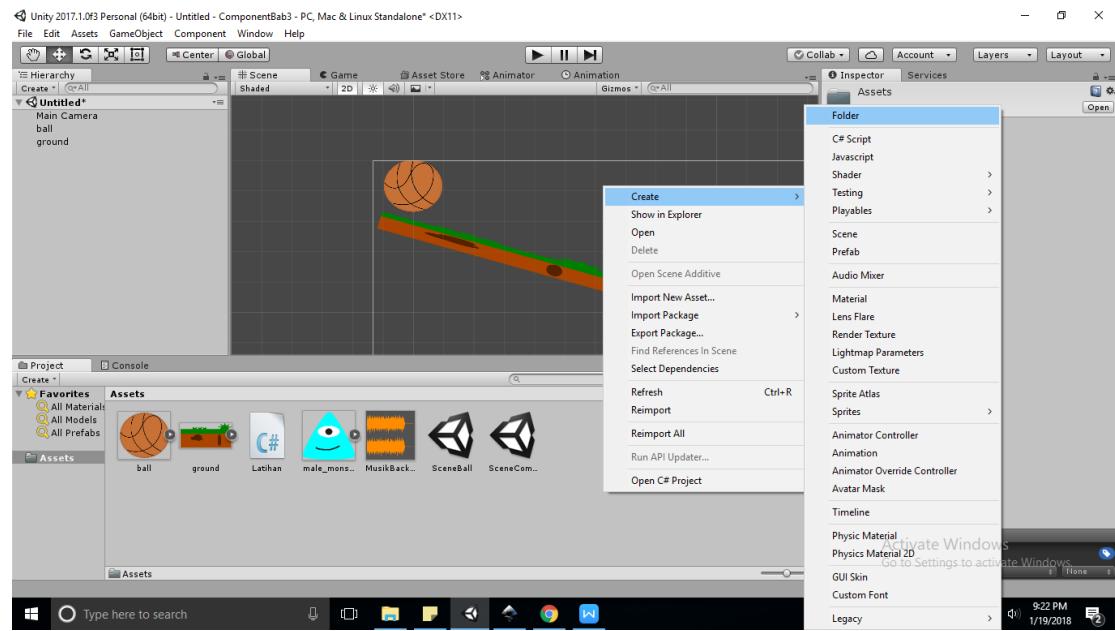
Tambahkan rigidbody2d static dan juga box collider 2d yang telah disesuaikan lalu rotasi terhadap sumbu z sehingga ground sekarang adalah sebagai berikut:



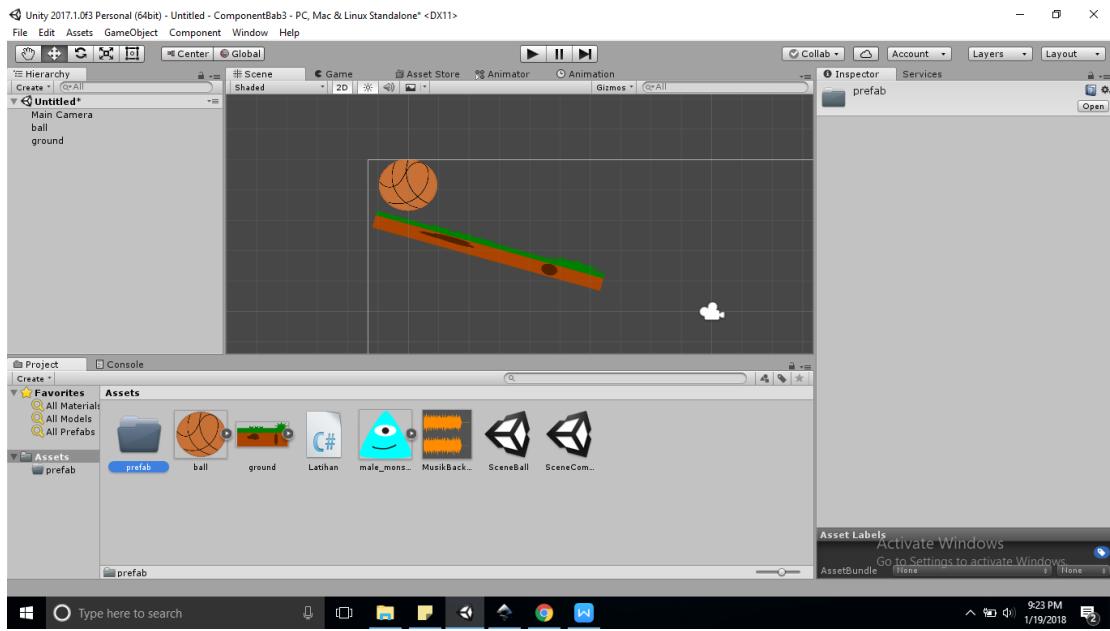
Silahkan dimainkan maka kita bisa melihat bola akan menggelinding lalu jatuh kebawah



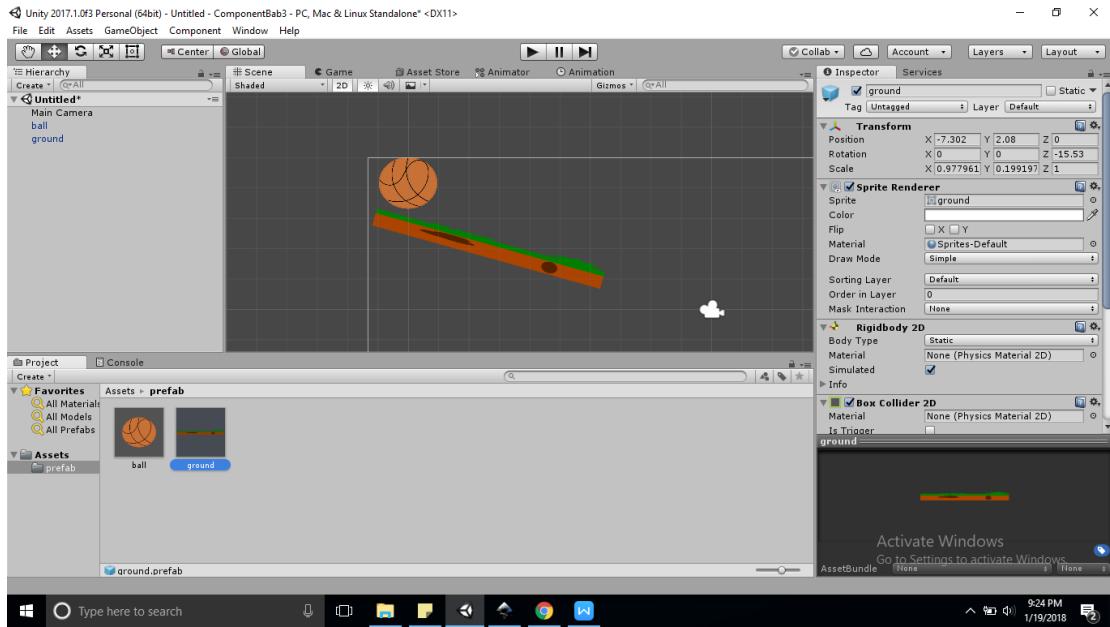
Sekarang kita jadikan ground dan ball kita sebagai prefab caranya adalah buat folder prefab dengan cara klik kanan lalu create folder sebagai berikut:



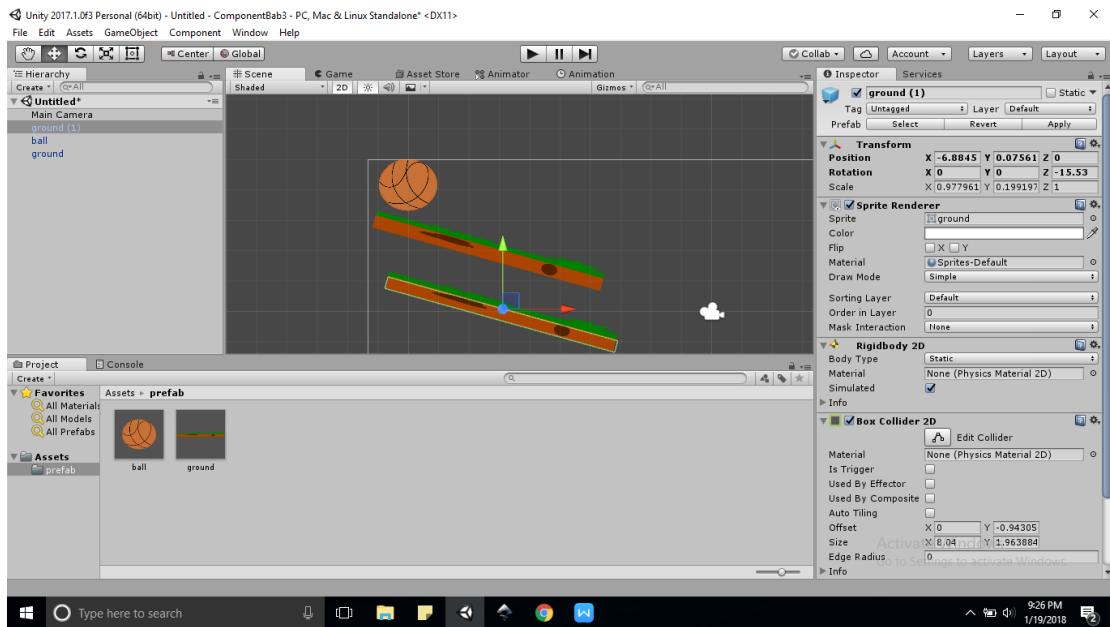
Beri nama prefab



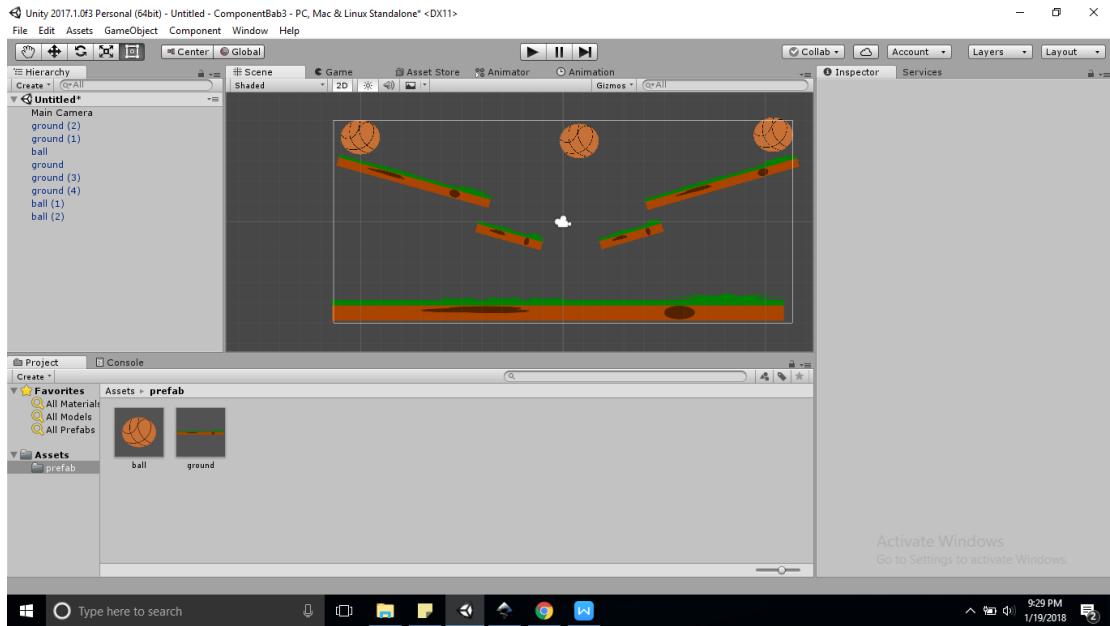
Setelah itu drag and drop gameobject ball dan ground kedalam folder prefab tersebut, ketika kita buka folder prefab maka didalamnya sudah ada prefab, perhatikan namanya sekarang ada extensi .prefab



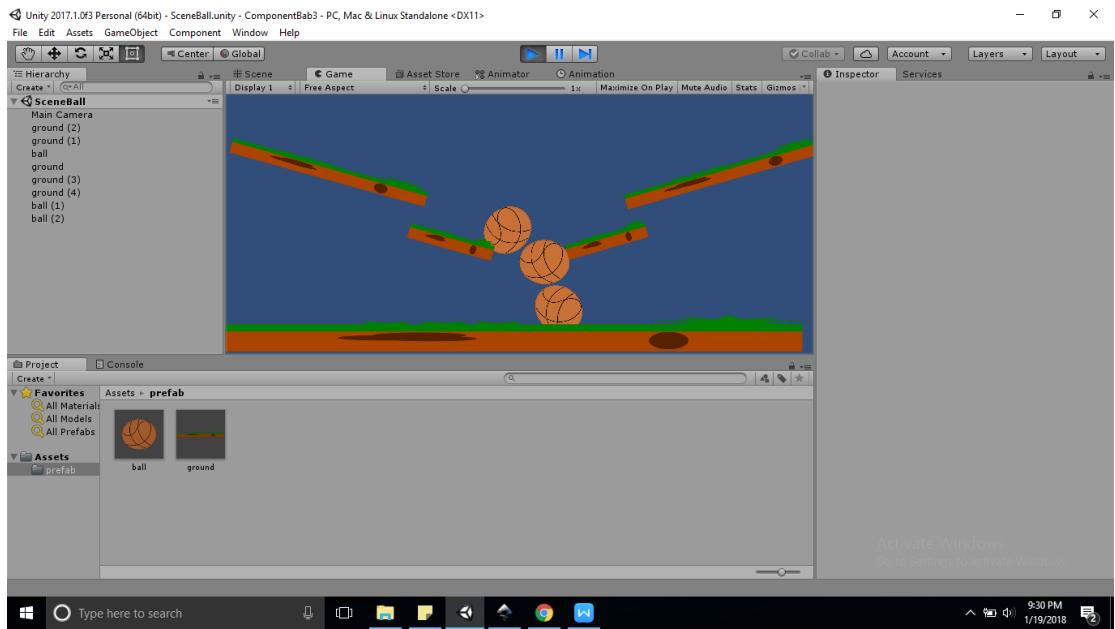
Kita bisa menggunakan prefab ini dengan cara drag and drop prefab tersebut ke scene dan jika diperhatikan maka saat di drag and drop prefab tersebut telah memiliki component dan pengaturan yang sama dengan gameobject awalnya.



Baik sekarang atur scene anda sebagai berikut

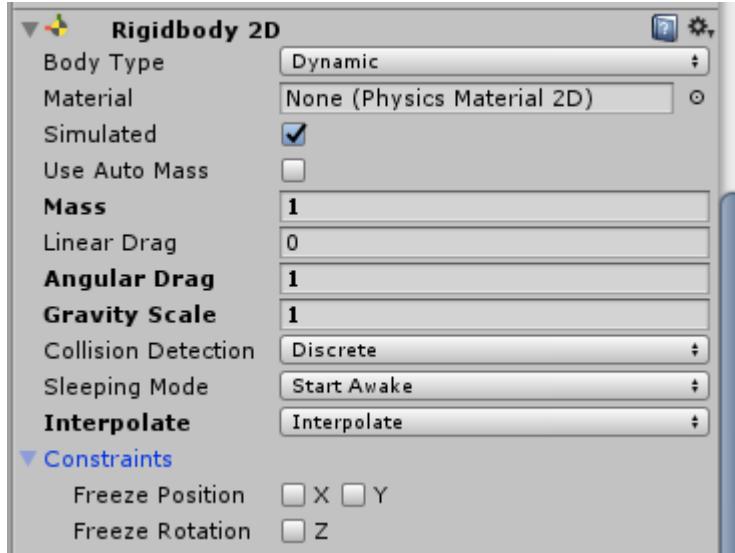


Silahkan mainkan



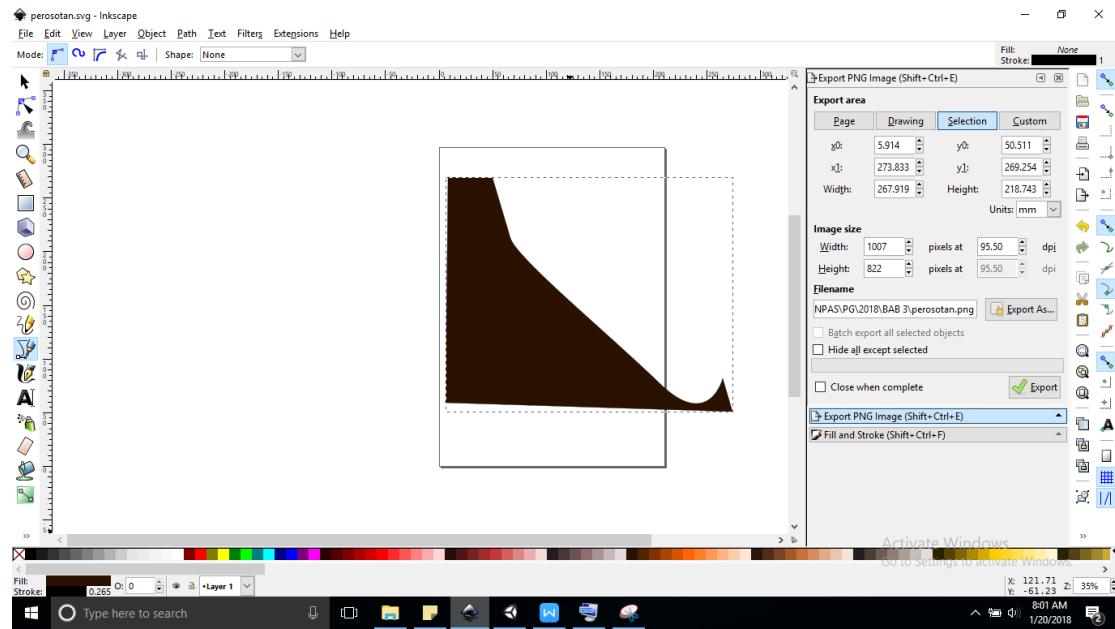
CONSTRAINTS

Berikutnya kita akan mempelajari mengenai constraints yang terdapat pada rigidbody2d.

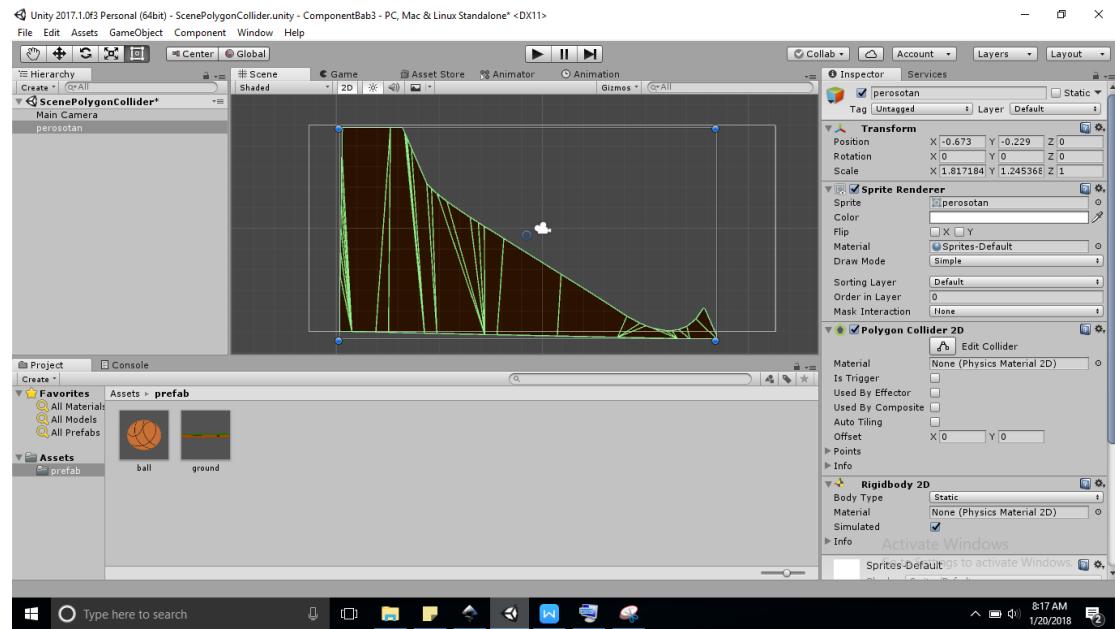


Fungsi daripada constraint adalah untuk mengunci posisi dan perputaran sebuah gameobject ketika physic pada rigidbodynya sedang bekerja. Untuk bisa memahami ini kita akan membuat bola agar menuruni sebuah tebing curam dan lalu melayang, disana kita akan mem-freeze rotation sehingga walaupun turun naik bola tidak akan berputar. Mari lakukanlah langkah berikut:

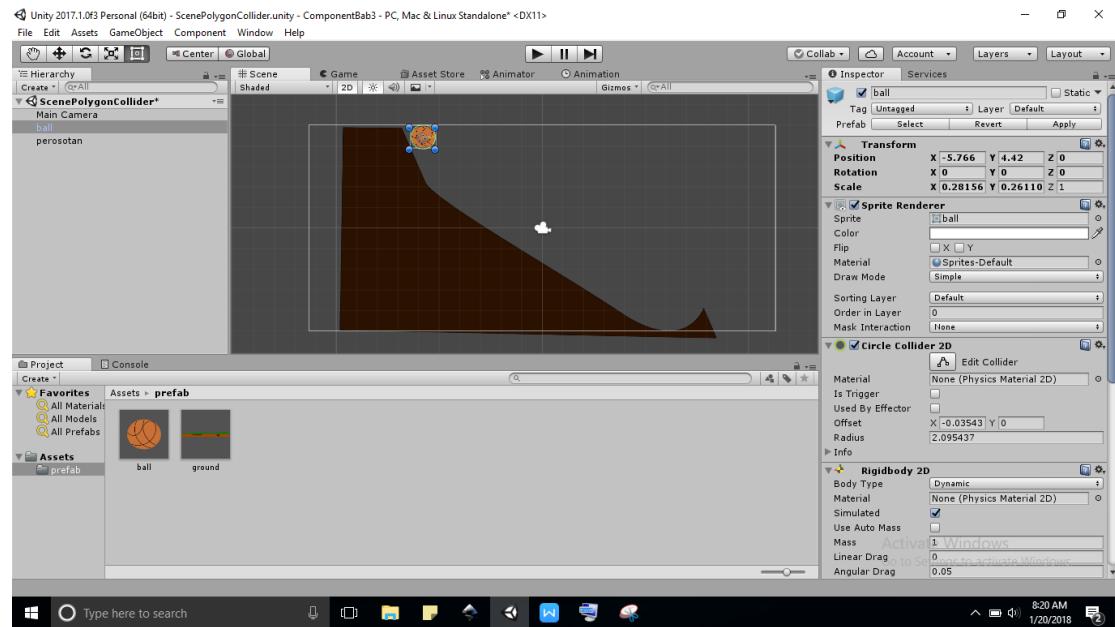
Pertama buatlah sebuah tebing atau perosotan di inkscape seperti berikut:



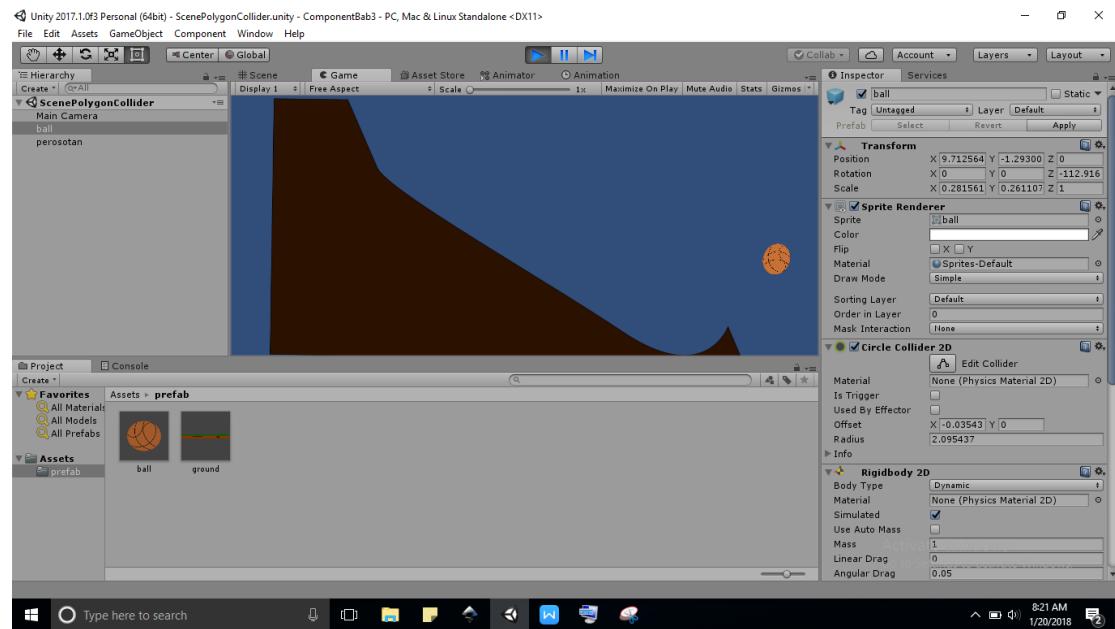
Export lalu simpan di sebuah scene baru, jadikan gameobject jangan lupa tambahkan rigidbodyd dan polygon collider 2d yang mampu membentuk collider dengan menyesuaikan bentuk gambar, hasilnya adalah sebagai berikut:



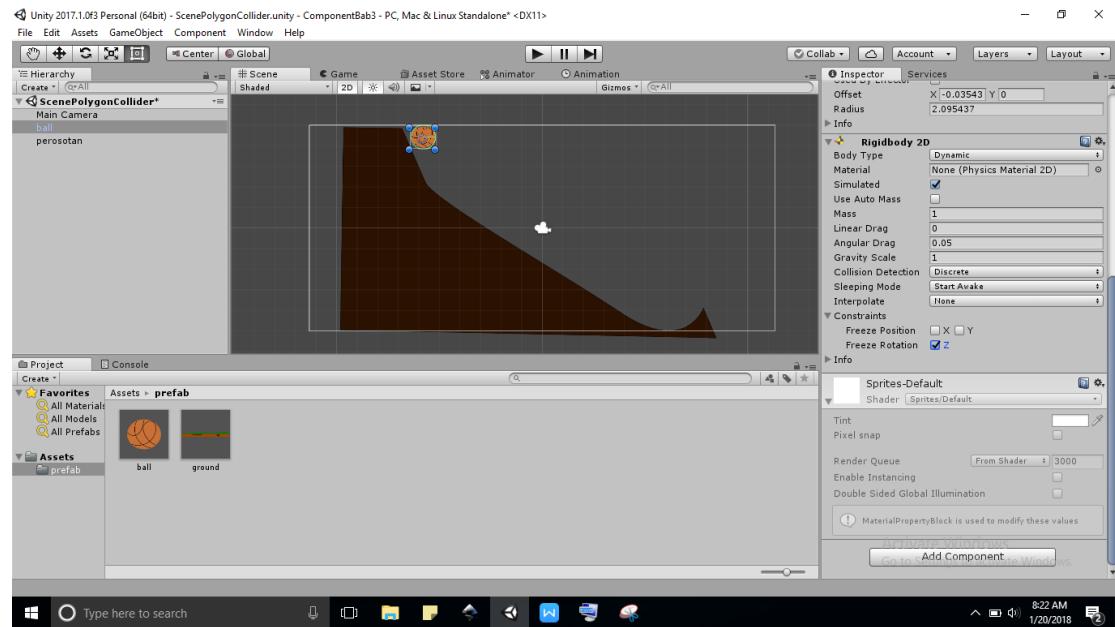
Tambahkan bola menggunakan prefab yang sudah ada. Ini membuktikan bahwa sekali kita membuat prefab maka bisa dipakai berkali-kali pada scene yang sama, scene yang berbeda-beda, bahkan proyek game yang berbeda, kedepan maksimalkan penggunaan prefab ini untuk mempercepat pengerjaan game anda. Baik sekarang bola tersebut posisikan diatas seperti ini:



Mainkan maka, kita bisa melihat bahwa bola menggelinding kebawah sambil berputar lalu terlempar keatas dan jatuh kebawah lagi sambil berputar. Seperti ini:



Jika kita ingin bola kita tidak berputar ketika kebawah maka freeze rotation sumbu z pada constraints, seperti ini:



Sekarang jika kita mainkan maka bola akan turun kebawah tapi tidak berputar. Silahkan dimainkan

