

Aplikasi *Binary Space Partitioning Tree* dalam Pembuatan Peta Acak pada *Role-Playing Game*

Vincent Christian Siregar - 13520136
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹author@itb.ac.id

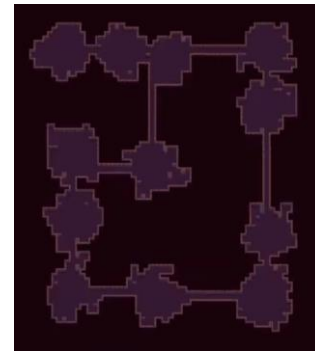
Role-Playing Game (RPG) adalah genre game yang disukai oleh banyak pemain di dunia. Game bertipe RPG disukai oleh berbagai kalangan mulai dari anak remaja sampai orang dewasa sekalipun. Keseruan dalam memainkan suatu peran di dunia virtual memberikan kesenangan tersendiri bagi pemainnya. Dunia virtual pada RPG memberi kesan seperti pemain sedang berada pada dunia lain. Beberapa game menerapkan fitur dimana dunia yang dibentuk bersifat random atau berbeda setiap game itu dijalankan. Makalah ini akan menjelaskan salah satu penerapan dari teori pohon yaitu *Binary Space Partitioning Tree* dalam pembuatan dunia atau peta acak (random).

Keywords— Binary Space Partitioning, BSP, Peta acak, RPG.

I. PENDAHULUAN

Role-Playing Game (RPG) adalah sebuah permainan yang pemainnya memainkan peran karakter dalam latar fiksi.[1] Terdapat beberapa sub-genre dari RPG yaitu Action RPG, MMORPG (Massively multiplayer online role-playing games, Roguelikes, Tactical RPG, Sandbox RPG, First-person party-based RPG, JRPG (Japanese Role-Playing Game), dan Monster Tamer. Dalam permainan RPG, biasanya pemain akan diceritakan sebagai sebuah karakter fiksi yang memiliki peran tertentu pada dunia tersebut. Karakter ini disesuaikan dengan latar atau tema dari game tersebut. Pemain disuguhkan dengan cerita yang menarik sehingga memberi kesan seru terhadap pemain.

Dalam permainan RPG, karakter pemain berada di sebuah dunia virtual (2D atau 3D) yang didesign sedemikian rupa seperti latar tempat cerita berlangsung. Pemain dapat melihat keseluruhan dunia dengan fitur peta atau *map* yang dapat memberi tahu pemain letak dan posisi dari suatu daerah. Peta utama atau tempat utama yang pemain tempati biasanya bersifat *fixed* atau ukuran dan isinya sudah ditentukan sesuai design dari *developer*. Namun, beberapa fitur dari RPG menggunakan fitur peta acak (*random map*) pada tempat-tempat tertentu. Salah satu contohnya adalah pembuatan *cave* atau *dungeon*. Biasanya player memiliki tujuan tertentu ketika memasuki *cave* atau *dungeon*. Hal ini disesuaikan dengan objektif game tersebut.



Gambar 1 Visualisasi cave pada RPG

Sumber:

https://www.youtube.com/watch?v=-OOCX6SVFsk&ab_channel=SunnyValleyStudio

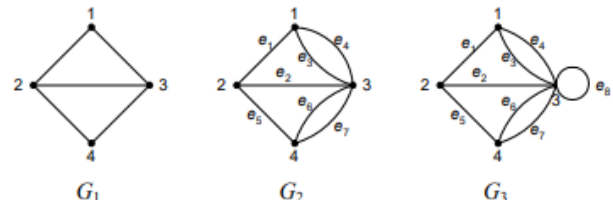
diakses pada tanggal 13 desember 2021

Salah satu algoritma yang dapat digunakan dalam membuat peta acak adalah menggunakan binary space partitioning tree. Algoritma ini digunakan agar pembagian wilayah peta terdistribusi dengan baik dan tidak menimpa wilayah lainnya.

II. LANDASAN TEORI

A. Graf

Graf adalah representasi objek – objek diskrit dan hubungan antara objek-objek tersebut. Graf dinyatakan dalam sebuah tupel (V,E). V (*Vertex*) merupakan himpunan tidak-kosong dari simpul-simpul. E (*edges*) merupakan himpunan sisi yang menghubungkan sepasang simpul.



Gambar 2. Contoh graf

Sumber :

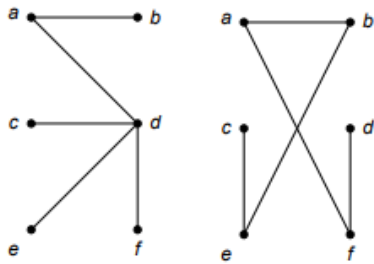
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

Diakses pada tanggal 13 desember 2021

B. Pohon (Tree)

Pohon didefinisikan sebagai graf tak-berarah yang tidak

mengandung sirkuit.

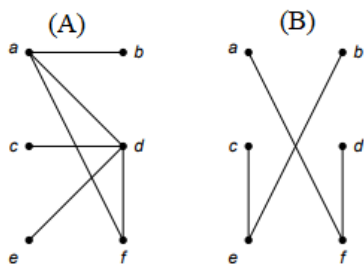


Gambar 3. Contoh graf yang termasuk pohon
Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>

Diakses pada tanggal 13 desember 2021

Perlu diingat bahwa tidak semua graf termasuk pohon. Graf yang memiliki sirkuit atau graf yang saling lepas tidak dapat dikategorikan sebagai pohon. Pada gambar di bawah, graf (A) memiliki sirkuit pada simpul a,d,c dan graf (B) merupakan graf saling lepas.



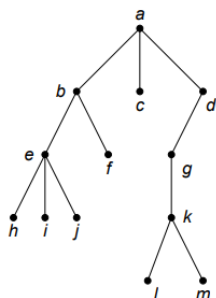
Gambar 4. Contoh graf yang tidak termasuk pohon
Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>

Diakses pada tanggal 13 desember 2021

C. Pohon Berakar (rooted tree)

Pohon berakar adalah pohon yang salah satu simpulnya merupakan akar dan sisinya diberi arah menjadi graf berarah.



Gambar 5. Pohon berakar
Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>

Diakses pada tanggal 13 desember 2021

Terdapat beberapa terminologi pada pohon berakar seperti

a. Anak (*child*) dan Orangtua (*parent*)

Anak adalah simpul yang berasal dari suatu simpul akar yang disebut sebagai orangtua. Contoh pada gambar 5, simpul a merupakan orangtua dari b,c, dan d dan simpul b,c, dan d adalah anak dari a.

b. Lintasan (*path*)

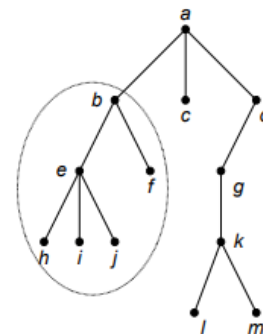
Lintasan adalah simpul yang dilewati untuk mencapai simpul tertentu. Contoh pada gambar 5, lintasan dari a ke j adalah a,b,e,j.

c. Saudara kandung (*sibling*)

Saudara kandung adalah simpul yang memiliki orangtua yang sama. Contoh pada gambar 5, e adalah anak kandung dari f tetapi e bukan saudara kandung dari g.

d. Upapohon (*subtree*)

Upapohon adalah pohon yang merupakan himpunan bagian dari suatu pohon.



Gambar 6. Upapohon dari pohon berakar
Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>

Diakses pada tanggal 13 desember 2021

e. Derajat (*degree*)

Derajat simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut. Contoh pada gambar 5, simpul a memiliki derajat tiga, simpul b memiliki derajat dua, dan simpul d memiliki derajat satu.

Derajat pohon itu sendiri adalah derajat maksimum simpul pohon tersebut

f. Daun (*leaf*)

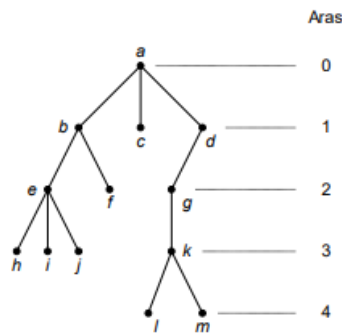
Daun adalah simpul yang tidak memiliki anak atau berderajat nol. Contoh pada gambar 5, h,i,j,l,m merupakan daun.

g. Simpul Dalam (*internal nodes*)

Simpul yang mempunyai anak disebut simpul dalam. Contoh pada gambar 5, simpul b, d, e, g, dan k adalah simpul dalam.

h. Aras (*level*)

Aras adalah tingkatan dari simpul suatu pohon. Aras ter atas dimulai dari 0.



Gambar 7. Aras dari pohon berakar

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>

Diakses pada tanggal 13 desember 2021

i. Kedalaman (*depth*)

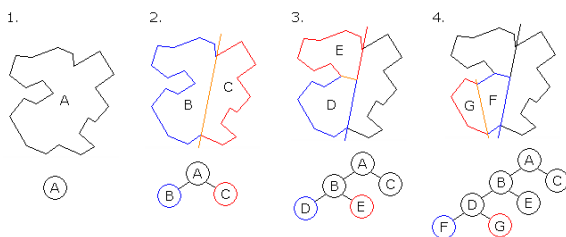
Kedalaman adalah aras maksimum dari suatu pohon. Contoh pada gambar 7, pohon memiliki kedalaman 4.

D. Binary Space Partitioning Tree (BSP)

BSP adalah sebuah metode untuk membagi ruang secara rekursif menjadi dua bagian menggunakan hiperbidang sebagai sekat pembagi.[3] Pembagian ruang ini dibentuk menggunakan struktur data pohon.

Metode BSP dapat digunakan dalam pembuatan peta atau map. Ruang yang digunakan dalam pembuatan di peta dibagi menjadi bagian-bagian yang di-generate secara random sehingga menghasilkan design peta yang unik setiap kali algoritma dijalankan.

Ruang disimpan dalam *daun* dari pohon. Banyaknya daun ditentukan dari ukuran ruangan dan ukuran minimal ruangan yang diinginkan.



Gambar 8. Visualisasi BSP

Sumber : id.wikipedia

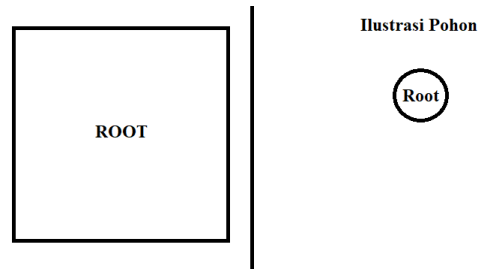
Diakses pada tanggal 13 desember 2021

III. PEMBUATAN PETA ACAK

A. Inisialisasi Akar (Root)

Tahapan pertama dalam pembuatan peta acak dengan BSP adalah menginisialisasi root dari pohon. Root pohon pertama adalah ukuran maksimal peta. Ukuran peta ditentukan oleh

pemrogram sesuai dengan ukuran peta yang diinginkan.

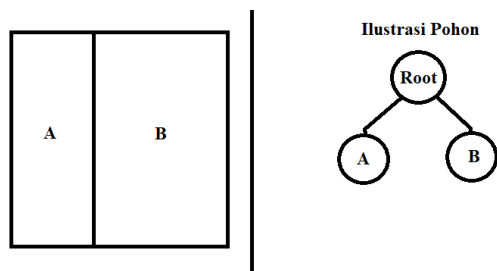


Gambar 9. Visualisasi root pada peta asli (kiri) dan ilustrasi pohon (kanan)

Sumber: Dokumentasi Pribadi

B. Pembelahan Akar (Splitting Root)

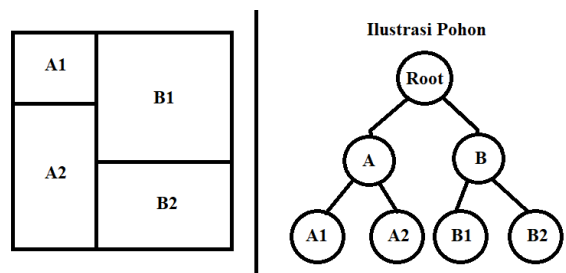
Root yang sudah diinisialisasi, dibagi menjadi dua bagian menjadi anak-anak dari root. Misalkan root dibagi menjadi A dan B (ilustrasi pada gambar 10). A dan B menjadi daun dari pohon. A dan B menjadi ruangan dari peta asli.



Gambar 10. Visualisasi root pada peta asli (kiri) dan ilustrasi pohon (kanan)

Sumber: Dokumentasi Pribadi

Simpul A dan simpul B dibagi menjadi 2 bagian lagi (ilustrasi pada gambar 11). Proses ini berulang sampai ruangan mencapai ukuran minimal yang ditentukan oleh pemrogram. Ketika mencapai ukuran minimal, simpul tersebut menjadi daun yang mencerminkan pembagian pembagian ruang final dari peta yang dibuat.



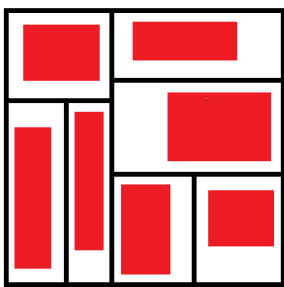
Gambar 11. Visualisasi root pada peta asli (kiri) dan ilustrasi pohon (kanan)

Sumber: Dokumentasi Pribadi

C. Pembuatan Ruang

Agar lebih realistis, ruangan yang dibentuk tidak persis berdempetan dengan ruangan lainnya. Seperti layaknya *cave* atau *dungeon* pada RPG, ruangan memiliki ukuran berbeda dan dihubungkan melalui lorong-lorong. Ruang direpresentasikan sebagai segiempat yang berada didalam pembagian ruangan.

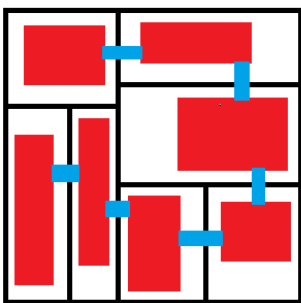
Setiap pembaligan ruangan memiliki satu buah ruangan (ilustrasi pada gambar 12).



Gambar 12. Visualisasi ruangan (merah)
Sumber: Dokumentasi Pribadi

D. Pembuatan Lorong

Lorong diperlukan untuk menyarukan ruangan yang sudah dibentuk pada poin C. Lorong menghubungkan ruangan satu dengan ruangan yang lain sehingga pemain dapat mengakses tiap ruangan dengan leluasa.



Gambar 13. Visualisasi ruangan (merah) dengan lorong (biru)
Sumber: Dokumentasi Pribadi

III. PENGENALAN UNITY DAN C#

A. Unity

Unity adalah *game engine* atau mesin permainan yang dikembangkan oleh Unity Technologies pertama kali pada bulan juni 2005 di Apple Inc. Unity awalnya merupakan mesin permainan eksklusif untuk Mac osx. Pada 2018, mesin permainan ini diperluas dan dapat mencakup hingga 25 platform. Unity menjadi mesin permainan yang populer dikalangan *game developer* atau pembuat game.

Unity dapat digunakan secara bebas dengan lisensi tertentu. Secara umum, setiap pengguna dapat menggunakan mesin permainan ini secara gratis untuk tujuan non komersial. Hal ini menjadi daya tarik tersendiri bagi *game developer* pemula karena tidak perlu mengeluarkan banyak budget untuk memulai membuat permainan.

Dalam pembuatan RPG diperlukan mesin permainan dalam merealisasikannya. Unity merupakan salah satu pilihan yang yang dapat digunakan oleh developer game. Unity dapat digunakan untuk membuat game 2D m(2 dimensi) ataupun game 3D (3 Dimensi).



Gambar 14. Contoh Permainan 2D
Sumber: <https://unity.com/how-to/difference-between-2D-and-3D-games>
Diakses pada tanggal 14 Desember 2021



Gambar 15. Contoh Permainan 3D
Sumber: <https://unity.com/how-to/difference-between-2D-and-3D-games>
Diakses pada tanggal 14 Desember 2021

B. C#

C# (# dibaca “sharp”) adalah bahasa pemrograman multi-paradigma. C# didesain oleh Anders Heljsberg dari microsoft pada tahun 2000. C# diakui sebagai international standard oleh Ecma pada tahun 2002 dan ISO tahun 2003.

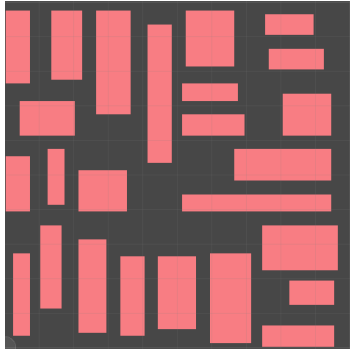
C# merupakan bahasa pemrograman yang berorientasi objek. Paradigma ini banyak dibutuhkan dalam pembuatan game terutama game yang menggunakan engine Unity.

C# digunakan sebagai script dalam *game engine* Unity. C# berfungsi sebagai kode yang menyimpan *behaviour* pada Unity. Pada pembahasan makalah ini, C# digunakan sebagai bahasa pemrograman untuk membentuk struktur data pohon yang kemudian digunakan untuk membuat peta acak pada game RPG

IV. VISUALISASI PETA ACAK DENGAN UNITY

Penerapan bsp dalam pembuatan peta acak dapat direalisasikan menggunakan bahasa pemrograman contohnya adalah bahasa pemrograman C#. Untuk menjalankan game, digunakan *game engine* unity yang dapat menggunakan *script* C# dalam pembuatan logic. Berikut adalah hasil visualisasi peta acak yang dihasilkan dari program.

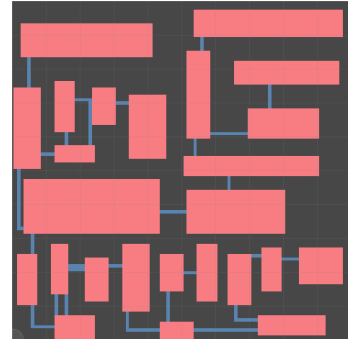
A. Pembuatan Peta tanpa Lorong



Gambar 16. Visualisasi ruangan (merah) dengan root 100x100 pixel, dengan ruangan minimal memiliki luas 10x10 pixel

Sumber: Adaptasi program dari

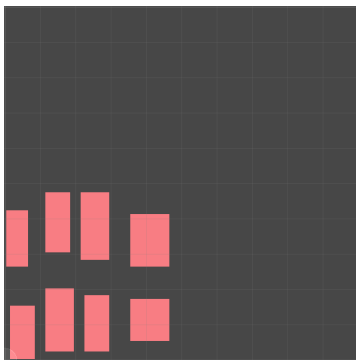
<http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021



Gambar 19. Visualisasi ruangan (merah) dan lorong (biru) dengan root 100x100 pixel, dengan ruangan minimal memiliki luas 10x10 pixel

Sumber: Adaptasi program dari

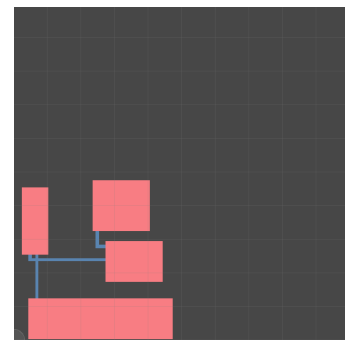
<http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021



Gambar 17. Visualisasi ruangan (merah) dengan root 50x50 pixel, dengan ruangan minimal memiliki luas 10x10 pixel

Sumber: Adaptasi program dari

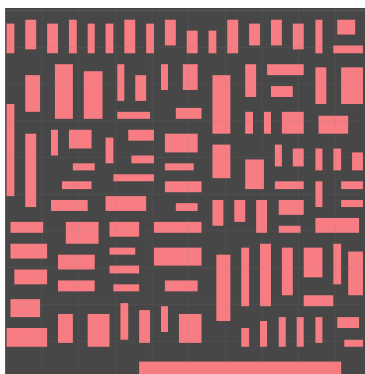
<http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021



Gambar 20. Visualisasi ruangan (merah) dan lorong (biru) dengan root 50x50 pixel, dengan ruangan minimal memiliki luas 10x10 pixel

Sumber: Adaptasi program dari

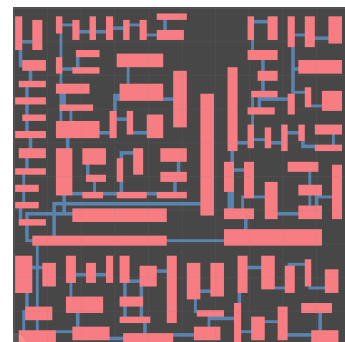
<http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021



Gambar 18. Visualisasi ruangan (merah) dengan root 100x100 pixel, dengan ruangan minimal memiliki luas 5x5 pixel

Sumber: Adaptasi program dari

<http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021



Gambar 21. Visualisasi ruangan (merah) dan lorong (biru) dengan root 100x100 pixel, dengan ruangan minimal memiliki luas 5x5 pixel

Sumber: Adaptasi program dari

<http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021

B. Pembuatan Peta dengan Lorong

V. SIMPULAN

BSP adalah salah satu metode dapat digunakan dalam membuat peta acak pada permainan RPG. Ruang yang dihasilkan tidak berdempatan ataupun saling menindih satu

sama lain karena pembagian ruang dilakukan dengan membelah suatu ruang menjadi dua sub ruang secara rekursif. Ukuran ruangan dan ukuran peta dapat disesuaikan oleh pemrogram sehingga peta yang dihasilkan sesuai dengan kemauan pemrogram.

VI. UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa karena rahmat dan karunia-Nya makalah ini dapat terselesaikan.

Terima kasih juga kepada dosen pengampu yang sudah senantiasa membimbing saya selama pembelajaran saya dalam mata kuliah IF 2120 Matematika Diskrit Institut Teknologi Bandung.

REFERENSI

- [1] Pat Harrigan, Noah Wardrip-Fruin (2010). *Second Person: Role-playing and Story in Games and Playable Media*. MIT Press.
- [2] https://www.youtube.com/watch?v=QOCX6SVFsk&ab_channel=SunnyValleyStudio diakses pada tanggal 13 desember 2021
- [3] Fan, X., Li, B. & Sisson, S.. (2018). The Binary Space Partitioning-Tree Process.
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf> Diakses pada tanggal 13 desember 2021
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf> Diakses pada tanggal 13 desember 2021
- [6] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf> Diakses pada tanggal 13 desember 2021
- [7] <https://unity.com/how-to/difference-between-2D-and-3D-games> diakses pada tanggal 14 Desember 2021
- [8] <http://www.rombdn.com/blog/2018/01/12/random-dungeon-bsp-unity/> diakses pada tanggal 14 desember 2021

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bekasi, 14 Desember 2020



Vincent Christian Siregar 13520136