

Perbandingan Pengambilan Rute Algoritma A* dan Hybrid A* dalam Proses Pencarian Rute pada Peta Dua Dimensi

Jonet Wira Murti 13518083
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518083@std.stei.itb.ac.id

Abstraksi—Algoritma telah digunakan dalam menyelesaikan berbagai persoalan komputasi yang ada, salah satunya persoalan pencarian rute. Dua algoritma yang dapat digunakan pada persoalan pencarian rute adalah A* dan hybrid A*. Kedua algoritma ini memiliki perbedaan cara pengambilan rute pada peta dua dimensi.

Kata Kunci—pencarian rute, algoritma A*, algoritma hybrid A*, perencanaan gerak, *nonholonomic*

I. PENDAHULUAN

Algoritma telah digunakan dalam menyelesaikan berbagai persoalan komputasi yang ada. Pemilihan algoritma untuk menyelesaikan persoalan-persoalan tersebut dilakukan agar hasil yang didapat dari algoritma yang dipilih merupakan hasil yang paling optimal atau sesuai dengan harapan. Salah satu persoalan pada bidang *engineering* yang membutuhkan algoritma sebagai dalam proses pemecahan masalahnya adalah persoalan pencarian rute.

Persoalan pencarian rute telah banyak ditemui dalam berbagai bidang *engineering* seperti pada bidang robotika, pembuatan kendaraan otomatis, permainan komputer, dan masih banyak lagi. Bidang-bidang tersebut tentu membutuhkan algoritma-algoritma pencarian rute yang optimal, dalam artian mampu mencari rute terbaik untuk dipilih berdasarkan batasan-batasan yang ada, dan mampu bekerja secara *real-time* dengan waktu komputasi yang cepat dan menggunakan sumber daya seminimal mungkin. Dua algoritma yang dapat digunakan pada persoalan-persoalan tersebut adalah algoritma A* dan hybrid A*.

Algoritma hybrid A* secara sederhana merupakan modifikasi dari algoritma A* biasa yang memungkinkan pencarian rute dengan hasil rute yang kontinu sehingga mampu diterapkan pada sistem yang *nonholonomic*, sehingga algoritma ini dapat digunakan pada pengembangan sistem-sistem seperti kendaraan otomatis atau sistem-sistem robotika lainnya.

II. LANDASAN TEORI

A. Peta Dua Dimensi

Peta dua dimensi atau ruang dua dimensi merupakan ruang yang direpresentasikan dengan dua koordinat cartesius, yaitu x dan y. Koordinat ini memungkinkan objek di dalamnya untuk melakukan gerak translasi pada sumbu x dan y serta gerak rotasi.

Pada makalah ini, digunakan peta dua dimensi berupa *grid map*, yaitu peta yang dipartisi menjadi persegi-persegi kecil berukuran $n \times n$ yang membentuk baris dan kolom. persegi-persegi ini nantinya akan merepresentasikan graf dengan setiap perseginya merupakan simpul dari graf tersebut.

	0	1	2	3	4
0					
1					
2					
3	S				
4					G

Gambar 1 Grid Map

(Sumber: Dokumen Penulis)

Pada peta tersebut akan terdapat dua jenis daerah, yaitu daerah kosong dan daerah penghalang (*obstacle*). Daerah kosong merupakan daerah yang dapat dengan bebas dilalui oleh objek, sedangkan daerah penghalang tidak dapat dilalui oleh objek. Jarak antara satu persegi/daerah ke persegi lain dapat dihitung menggunakan jarak *euclidean* untuk ruang dua dimensi dengan formula :

$$\text{Jarak Euclidean} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

B. Algoritma A*

Algoritma A* atau A-bintang merupakan algoritma pencarian rute yang digunakan untuk mencari rute terpendek dari suatu simpul asal ke suatu simpul tujuan pada sebuah graf. Algoritma ini tergolong kedalam kategori *informed search*, yaitu algoritma-algoritma pencarian yang menggunakan informasi atau pengetahuan tambahan mengenai ruang pencarian ketika melakukan proses pencarian. Informasi tambahan ini merupakan sebuah fungsi heuristik.

Pada implementasinya, algoritma ini akan melakukan pemilihan simpul yang akan diekspansi dari kumpulan simpul hidup dengan menghitung fungsi $f(n) = g(n) + h(n)$ dan memilih simpul dengan nilai $f(n)$ paling kecil diantara simpul-simpul hidup lainnya. $g(n)$ merupakan fungsi yang menghitung total jarak yang telah dilalui sampai saat ini, yaitu jarak simpul asal ke simpul $n-1$ ditambah jarak simpul yang sedang diekspansi (n) ke simpul $n-1$. Jarak ini dihitung menggunakan jarak *euclidean*. Fungsi $h(n)$ merupakan fungsi heuristik yang juga dihitung menggunakan jarak *euclidean* antara simpul yang sedang diekspansi dengan simpul tujuan.

Berikut ini merupakan *pseudocode* dari algoritma A*:

```

1 Put node_start in the OPEN list with  $f(\text{node\_start}) = h(\text{node\_start})$  (initialization)
2 while the OPEN list is not empty {
3   Take from the open list the node node_current with the lowest
4    $f(\text{node\_current}) = g(\text{node\_current}) + h(\text{node\_current})$ 
5   if node_current is node_goal we have found the solution; break
6   Generate each state node_successor that come after node_current
7   for each node_successor of node_current {
8     Set successor_current_cost =  $g(\text{node\_current}) + w(\text{node\_current}, \text{node\_successor})$ 
9     if node_successor is in the OPEN list {
10      if  $g(\text{node\_successor}) \leq \text{successor\_current\_cost}$  continue (to line 20)
11     } else if node_successor is in the CLOSED list {
12      if  $g(\text{node\_successor}) \leq \text{successor\_current\_cost}$  continue (to line 20)
13     } Move node_successor from the CLOSED list to the OPEN list
14     } else {
15      Add node_successor to the OPEN list
16      Set  $h(\text{node\_successor})$  to be the heuristic distance to node_goal
17     }
18     Set  $g(\text{node\_successor}) = \text{successor\_current\_cost}$ 
19     Set the parent of node_successor to node_current
20   }
21   Add node_current to the CLOSED list
22 }
23 if (node_current != node_goal) exit with error (the OPEN list is empty)

```

Gambar 2 A* Pseudocode

(Sumber: <http://mat.uab.cat/~alseda/MasterOpt/AStar-Algorithm>)

C. Algoritma Hybrid A*

Cara kerja algoritma *hybrid A** tidak berbeda jauh dengan algoritma A* biasa. Perbedaan mendasar antara kedua algoritma ini adalah pada konfigurasi ruang tempat pencarian rute. Pada algoritma A*, objek yang melakukan pencarian merupakan objek titik yang tidak memiliki dimensi ruang, sehingga objek dapat bergerak kemanapun pada peta. Namun, pada dunia nyata, objek seharusnya bisa melakukan rotasi. Sehingga, pada algoritma *hybrid A**, objek yang melakukan

pencarian merupakan objek yang memiliki posisi dan orientasi. Pada makalah ini, digunakan objek dua dimensi yang memiliki panjang dan lebar. Sehingga, *state* dari objek pada setiap simpul tidak hanya ditandai dengan posisi objek tersebut, melainkan orientasi dari objek ketika berada diatas sebuah kotak pada peta. Maka, sebuah objek yang berada di atas sebuah simpul memiliki *state* $n(x, y, \theta)$, dimana x menyatakan posisi di koordinat x , y menyatakan posisi di koordinat y , dan θ menyatakan orientasi objek di simpul tersebut.

Perhitungan posisi pada algoritma ini menghasilkan nilai yang kontinu, karena terdapat perhitungan rotasi objek selain gerak translasinya. Namun, karena algoritma A* yang diterapkan memanfaatkan peta dengan *grid* sebagai representasi grafnya, maka nilai tersebut harus dikonversi menjadi nilai diskrit dengan formula :

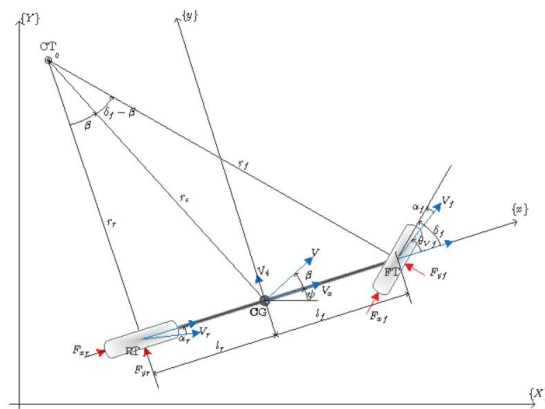
$$\text{Posisi_diskrit} = \text{floor}((\text{Posisi_kontinu} - \text{titik_asal})/\text{lebar_kotak})$$

Dalam hal ini digunakan titik asal (0, 0) dan peta dengan lebar kotak 1, sehingga formula hanya menjadi:

$$\text{Posisi_diskrit} = \text{floor}(\text{Posisi_kontinu})$$

D. Single-track Vehicle Model

Pada algoritma *hybrid A** yang diterapkan dalam makalah ini, objek yang melakukan pencarian rute dapat diasumsikan sebagai *Single-track Vehicle Model*, yaitu model kendaraan yang menggunakan dua roda sebagai penggerak. Hal tersebut dipilih untuk memudahkan proses perhitungan pergerakan objek.



Gambar 3 Single-track Vehicle Model

(Sumber:

<https://www.semanticscholar.org/paper/Lane-estimation-using-a-vehicle-kinematic-lateral-Kang-Lee/0240900e8f49485e164e848e71a8bf07b8cdcc19/figure/0>

Dalam menghitung pergerakan kendaraan *Single-track*, dibutuhkan peubah-peubah berupa posisi objek pada koordinat dua dimensi, orientasi objek pada koordinat dua dimensi, jarak antar roda, sudut belok roda, dan panjang perpindahan kendaraan. Peubah-peubah tersebut digunakan untuk mencari posisi dan orientasi akhir dari objek setelah melakukan gerakan.

III. ANALISIS DAN PEMBAHASAN

Bagian tiga dari makalah akan berfokus pada proses implementasi, perbandingan, dan pengujian dari algoritma A* dan *hybrid A**. Pada proses implementasi terdapat beberapa asumsi yang dilakukan terutama pada implementasi algoritma *hybrid A** dengan menghilangkan beberapa besaran fisis yang tidak diperlukan atau pembatasan terhadap nilai dari suatu peubah.

A. Struktur Data

Algoritma A* dan *hybrid A** memiliki perbedaan struktur data pada simpul graf peta.

```
class Node:
    def __init__(self, position, goal_position):
        self.position = [position[0],
            position[1]]
        self.parent = None
        self.h = euclidean_dist(self.position,
            goal_position)
        self.g = 0
```

Pada algoritma A* biasa, sebuah simpul graf hasil pembangkitan menyimpan informasi berupa posisi simpul tersebut pada peta (berupa koordinat x dan y), nilai fungsi heuristik h untuk simpul tersebut, *variable* Node yang menunjuk ke simpul sebelum simpul tersebut, dan nilai fungsi g yang diinisialisasi dengan 0. Nilai fungsi g akan dihitung pada proses algoritma.

```
class Hybrid_node:
    def __init__(self, position, theta,
        goal_position):
        self.dis_position =
            [math.floor(position[0]),
            math.floor(position[1])]
        self.con_position = [position[0],
            position[1]]
```

```
self.theta = theta
self.parent = None
self.h =
    euclidean_dist(self.dis_position, goal_position)
self.g = 0
```

Pada algoritma *hybrid A** terdapat penambahan atribut theta pada kelas Node yang telah dibuat sebelumnya. Atribut ini menyatakan orientasi objek pada sebuah simpul yang dibangkitkan. Pada proses implementasi yang dilakukan terdapat pembatasan dari orientasi yang mungkin untuk objek pada simpul menjadi 4 jenis orientasi, yaitu 0, 1, 2, dan 3 yang masing-masing merepresentasikan sudut kelipatan 90 derajat. Kemudian, terdapat dua jenis posisi, yaitu posisi diskrit dan posisi kontinyu. Posisi diskrit merupakan posisi tiap simpul atau persegi pada peta yang akan digunakan pada proses pembangkitan simpul pada algoritma, sedangkan posisi kontinyu merupakan posisi sebenarnya dari objek pada peta yang akan digunakan pada proses pergerakan dan rekonstruksi rute di akhir algoritma. Namun karena perhitungan trigonometri dari orientasi objek menghasilkan nilai yang diskrit (sudut kelipatan 90), maka posisi kontinyu pada implementasi ini sama dengan posisi diskrit.

B. Satuan Gerak

Algoritma A* memiliki delapan satuan gerak untuk objek yang berada pada simpul, yang masing-masing melakukan gerak ke daerah atau persegi yang berada di sekelilingnya, baik itu secara horizontal, vertikal, maupun diagonal. Sehingga setiap jenis gerak dapat direpresentasikan sebagai *tuple* dengan aritas 2 dengan masing-masing elemen menggambarkan Berikut ini merupakan implementasi satuan gerak pada algoritma A* biasa.

```
astar_move = [[0, 1], [0, -1], [1, 0], [-1, 0],
    [-1, 1], [1, 1], [-1, -1], [1, -1]]
```

Pada algoritma *hybrid A**, pergerakan objek lebih terbatas karena terdapat berbagai *constraint* terkait objek pada simpul. Pada implementasi yang dilakukan terdapat beberapa asumsi yang dilakukan penulis. Asumsi yang pertama adalah objek hanya dapat melakukan gerak maju. Kemudian, implementasi juga mengabaikan peubah-peubah seperti sudut belok roda dan ukuran objek, sehingga sudut belok total kendaraan telah didefinisikan dari awal menjadi tiga jenis, yaitu 90 derajat ke kiri, 90 derajat ke kanan, dan 0 derajat atau gerakan maju, sehingga posisi dan orientasi akhir dari objek dapat di enumerasi menjadi bilangan-bilangan diskrit. Sehingga satu jenis gerak dapat direpresentasikan sebagai *tuple* dengan aritas 2 dengan elemen pertama merupakan perintah untuk maju dan elemen kedua merupakan arah gerak (maju, kanan, kiri). Berikut ini merupakan implementasi satuan gerak pada algoritma *hybrid A**.

```
hybrid_move = [[1, 0], [1, -1], [1, 1]]
```

Gambar 6 Visualisasi Rute A* Kasus Uji 1

(Sumber: Dokumen Penulis)

C. Simulasi Pencarian Rute

Pada proses visualisasi, titik mulai ditandai dengan lingkaran berwarna biru, sedangkan titik akhir ditandai dengan lingkaran berwarna merah.

1. Kasus Uji 1

	0	1	2	3
0				
1				
2	S			G
3				

Gambar 4 Kasus Uji 1

(Sumber: Dokumen Penulis)

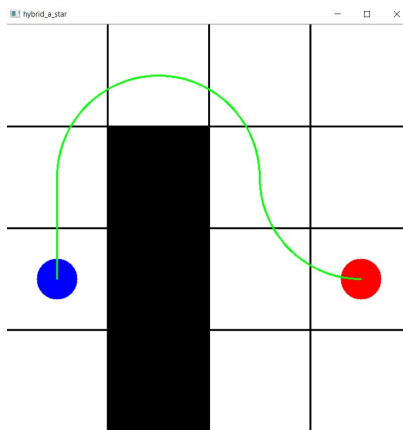
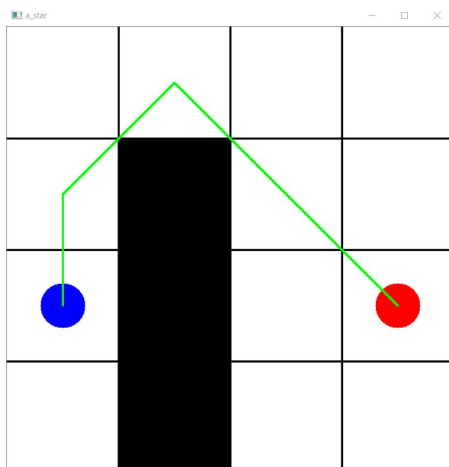
a) Algoritma A*

Berikut ini merupakan Hasil perhitungan rute dan visualisasi dari algoritma A* biasa untuk kasus uji 1.

rute yang dihasilkan A* :
[[2, 0], [1, 0], [0, 1], [1, 2], [2, 3]]

Gambar 5 Hasil Perhitungan Rute A* Kasus Uji 1

(Sumber: Dokumen Penulis)



Gambar 8 Visualisasi Rute hybrid A* Kasus Uji 1

(Sumber: Dokumen Penulis)

2. Kasus Uji 2

	0	1	2	3
0				G
1				
2	S			
3				

Gambar 9 Kasus Uji 2

(Sumber: Dokumen Penulis)

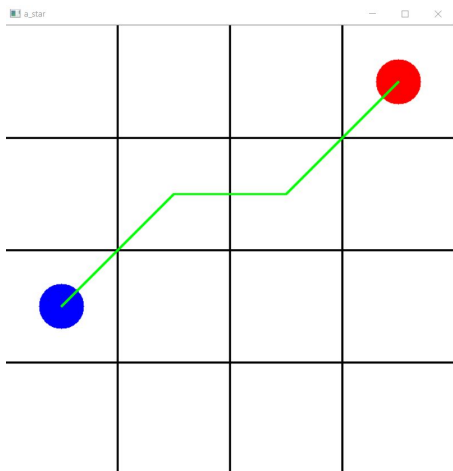
a) Algoritma A*

Berikut ini merupakan Hasil perhitungan rute dan visualisasi dari algoritma A* biasa untuk kasus uji 2.

```
rute yang dihasilkan A* :
[[2, 0], [1, 1], [1, 2], [0, 3]]
```

Gambar 10 Hasil Perhitungan Rute A* Kasus Uji 2

(Sumber: Dokumen Penulis)



Gambar 11 Visualisasi Rute A* Kasus Uji 2

(Sumber: Dokumen Penulis)

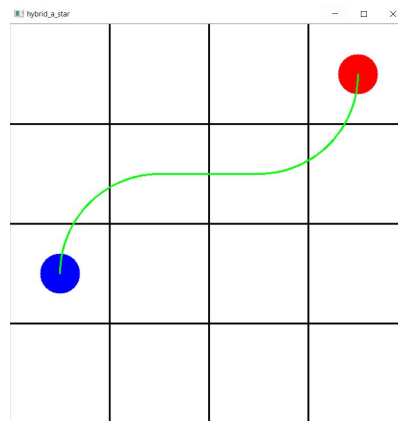
b) Algoritma *Hybrid A**

Berikut ini merupakan Hasil perhitungan rute dan visualisasi dari algoritma *hybrid A** untuk kasus uji 2. Pada kasus uji 2 terdapat dua percobaan untuk algoritma *hybrid A**, yaitu percobaan untuk objek dengan orientasi awal menghadap ke arah atas atau utara dan percobaan untuk objek dengan orientasi awal menghadap ke arah kanan atau timur.

```
rute yang dihasilkan Hybrid A* :
[[2, 0], [1, 1], [1, 2], [0, 3]]
```

Gambar 12 Hasil Perhitungan Rute *hybrid A** Kasus Uji 2 dengan orientasi awal ke arah utara

(Sumber: Dokumen Penulis)



Gambar 13 Visualisasi Rute *hybrid A** Kasus Uji 2 dengan orientasi awal ke arah utara

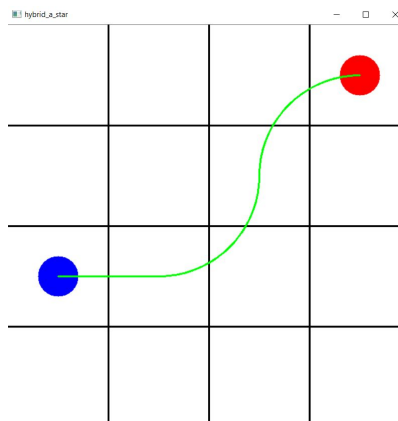
(Sumber: Dokumen Penulis)

Berikut ini merupakan percobaan kedua pada algoritma *hybrid A** pada kasus uji nomor 2 dengan orientasi awal ke arah timur.

```
rute yang dihasilkan Hybrid A* :
[[2, 0], [2, 1], [1, 2], [0, 3]]
```

Gambar 14 Hasil Perhitungan Rute *hybrid A** Kasus Uji 2 dengan orientasi awal ke arah timur

(Sumber: Dokumen Penulis)



Gambar 15 Visualisasi Rute *hybrid A** Kasus Uji 2 dengan orientasi awal ke arah timur

(Sumber: Dokumen Penulis)

D. Analisis Hasil Simulasi

Dari 2 kasus uji yang dilakukan pada algoritma A* dan *hybrid A** dapat dilihat bahwa rute yang dihasilkan antara kedua algoritma ini memiliki bentuk yang cukup berbeda. algoritma *hybrid A** menghasilkan bentuk rute yang kontinyu

dibandingkan dengan algoritma A* biasa. Jika meninjau dari proses pemilihan simpul yang dijadikan sebagai pilihan rute, kasus uji 1 menghasilkan urutan simpul yang sama antara kedua algoritma tersebut. Namun, pada kasus uji 2, terdapat perbedaan pemilihan simpul bahkan pada algoritma *hybrid A** itu sendiri. Hal ini karena kondisi atau *state* dari objek sangat dipengaruhi oleh orientasi awal objek. Pembentukan rute yang lebih kontinu pada algoritma *hybrid A** membuat algoritma tersebut cocok untuk diterapkan pada sistem yang *nonholonomic* seperti pada kendaraan. Namun, karena orientasi awal sangat mempengaruhi langkah selanjutnya pada algoritma *hybrid A**, maka kesalahan orientasi awal dapat berakibat pada tidak ditemukannya jalur untuk mencapai kondisi akhir.

Pengembangan lebih lanjut pada implementasi algoritma *hybrid A** dapat dilakukan dengan memperhitungkan peubah-peubah yang diabaikan sebelumnya, baik yang bersifat internal (terdapat dalam sistem itu sendiri) atau eksternal (berasal dari lingkungan). Kemudian, hal lain yang juga dapat dikembangkan adalah pada perhitungan fungsi evaluasi dalam pembangkitan simpul dengan mengubah fungsi $g(n)$ dan $h(n)$. Hal tersebut dapat dilakukan dengan menambahkan status-status lain dari objek atau sistem yang sedang bergerak, seperti banyak energi atau bahan bakar yang dimiliki serta pengaruh objek-objek di sekitarnya.

IV. KESIMPULAN

Dari simulasi yang telah dilakukan dengan mengabaikan beberapa peubah, Algoritma A* dan *hybrid A** memiliki perbedaan bentuk rute yang diambil. Algoritma A* memiliki bentuk rute yang patah-patah sedangkan algoritma *hybrid A** memiliki rute yang lebih kontinu sehingga membuat algoritma *hybrid A** cukup baik untuk diterapkan pada pencarian rute untuk objek yang memiliki dimensi ruang dan keterbatasan gerak.

TAUTAN VIDEO YOUTUBE

Berikut ini merupakan link video makalah pada youtube https://youtu.be/ISF_zPXxR5o

UCAPAN TERIMA KASIH

Puji syukur kehadirat Allah SWT, atas berkah dan rahmat-Nya, sehingga penulis dapat menyelesaikan makalah berjudul “Perbandingan Pengambilan Rute Algoritma A* dan *Hybrid A** dalam Proses Pencarian Rute pada Peta Dua Dimensi”. Penulis mengucapkan terima kasih kepada Dr. Nur Ulfa Maulidevi ST, M.Sc. selaku dosen pengajar IF2211 Strategi Algoritma, yang telah memberikan bimbingan dalam mata kuliah Strategi Algoritma ini. Terima kasih juga penulis ucapkan kepada keluarga dan teman-teman penulis yang selalu memberikan dukungan kepada penulis.

REFERENSI

- [1] <https://pdfs.semanticscholar.org/6e00/16024b257040db590d2de352556f64f46787.pdf> diakses pada 2 Mei 2020
- [2] <https://blog.habrador.com/2015/11/explaining-hybrid-star-pathfinding.html> diakses pada 2 Mei 2020
- [3] https://commonroad.in.tum.de/static/docs/vehicleModels_commonRoad.pdf diakses pada 2 Mei 2020
- [4] https://ai.stanford.edu/~ddolgov/papers/dolgov_gpp_stair08.pdf diakses pada 2 Mei 2020

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.

Bandung, 04 Mei 2020



Nama dan NIM