Pemanfaatan vektor dalam pendeteksian gerakan melingkar pada layar sentuh

Candra Ramsi 13514090
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*candra_ramsi@students.itb.ac.id*

***Abstract*—Layar sentuh sudah banyak diadaptasi pada banyak perangkat. Hal ini disebabkan maraknya penggunaan *smartphone* yang pada umumnya memiliki fitur layar sentuh. Penggunaan layar sentuh untuk navigasi sudah sangat baik tetapi penggunaan layar sentuh untuk permainan komputer sering disalahgunakan. Tetapi, ada beberapa permainan yang memanfaatkan layar sentuh sepenuhnya. Layar sentuh memiliki hal yang mirip dan berbeda dengan media mouse. Layar sentuh memberikan cara lain untuk bernavigasi dan bermain. Cara-cara ini memerlukan algoritma untuk mendeteksinya. Algoritma yang akan dibahas pada makalah ini adalah algoritma pendeteksi gerak melingkar pada layar sentuh.**

***Keywords*—layar sentuh, touch gesture, lingkaran.**

I. LATAR BELAKANG

Pengunaan fitur layar sentuh pada permainan komputer sudah marak pada beberapa tahun terakhir. Hal ini disebabkan oleh meledaknya jumlah pengguna *smartphone* dan menurunnya pengguna telepon genggam. *Smartphone* pada umumnya memiliki fitur layar sentuh.

Pengembangan permainan komputer untuk media baru ini pun sudah banyak dilakukan. Tiap harinya, lima ratus permainan baru dirilis pada platform android[1]. Permainan-permainan tersebut sangat beragam *genre*-nya. Salah satu genre yang paling banyak dimainakan adalah permainan Turn based RPG.

Penggunaan fitur layar sentuh pada tiap-tiap *genre* berbeda-beda. Beberapa *genre* permainan memilih untuk mengimplementasikan tombol virtual yang digunakan sama seperti layaknya permainan console dimainkan dengan game controller. Beberapa genre lainnya menggunakan layar sebagai satu tombol besar saja. Beberapa genre memanfaatkan gerakan-gerakan yang biasanya digunakan untuk bernavigasi pada layar sentuh.

Pada contoh-contoh diatas terdapat berbagai cara pengembang permainan mobile memanfaatkan layar sentuh dan beberapa tidak memanfaatkannya. Pengunaan tombol virtual pada layar sentuh tidak memanfaatkan sama sekali fitur layar sentuh. Tetapi hal ini bukan merupakan kesalahan pengembang permainan.

Media layar sentuh untuk permainan komputer memang tidak sesuai untuk semua *genre* game. Permainan-permainan yang membutuhkan berbagai macam tombol dan waktu reaksi yang ecpat untuk dimainkan tidak cocok untuk media layar sentuh ini. Contoh permainan tersebut adalah permainan dengan *genre* RTS (Real time strategy) dan FPS (First person shooter). [2]

Gambar 1. Permainan dengan *genre RTS*[7]

Namun, ada beberap genre permainan yang sangat sesuai dengan media layar sentuh ini. Salah satu contoh yang baik adalah fruit ninja. Fruit ninja menfaatkan gerakan swipe dengan sangat baik. Bahkan cara bernavigasinya pun menggunakan gerakan swipe.

Gambar 2. Fruit ninja – Permainan yang memanfaatkan layar sentuh dengan baik [6]

II. LATAR BELAKANG MASALAH

Layar sentuh merupakan media yang relatif baru. Layar sentuh mempunyai banyak kemiripan dengan *mouse*. Pada dasarnya, menyentuh tidak berbeda dengan mengarahkan dan memencet. Namun, layar sentuh memiliki perbedaan yaitu fitur *multitouch* yang memperbolehkan layar sentuh untuk menerima berbagai sentuhan dalam sebuah layar.

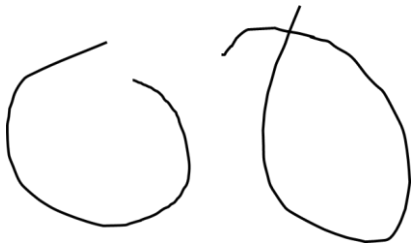
Hal ini memberikan cara lain untuk bernavigasi. Berbagai gerakan sudah dibuat untuk mempermudah navigasi pada layar sentuh. Gerak-gerakan ini terdiri dari *tap, double tap, press, rotate, pinch, spread, drag, swipe,* dan *flick*. Gerakan-gerakan ini sangat beragam dan cukup intuitif untuk navigasi pada layar sentuh.[3]

Permainan berbeda dengan applikasi lainnya dapat memanfaatkan gerakan lain selain gerakan-gerakan standar. Gerakan-gerakan yang tidak umu digunakan pada navigasi adalah gerakan-gerakan membentuk lingkarang, membentuk zigzag, d.s.b.

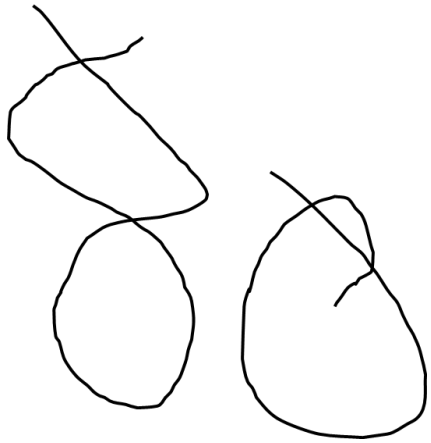
Pendeteksian gerakan-gerakan pada layar sentuh bukan merupakan hal yang mudah. Implementasi pendeteksian gerakan-gerakan standar sudah banyak dan mudah untuk diimplementasikan pada applikasi. Tetapi, gerakan-gerakan yang tidak umum harus diimplementasikan. Makalah ini mencoba menjawab pertanyaan bagaimana cara mendeteksi gerakan melingkar pada layar sentuh.

III. DEFINISI LINGKARAN

Gerakan melingkar yang dimaksud pada masalah adalah gerakan membuat lingkaran. Lingkaran yang dimaksud pada masalah tidak harus merupakan lingkaran yang sempurna. Lingkaran dapat berbentuk oval. Lingkaran yang terbentuk juga tidak harus simetris. Awal dan akhir dari pengambaran lingkaran tidak harus bersentuhan namun harus mendekati jika tidak bersentuhan. Mendekati pada definisi diatas adalah tidak lebih dari serratus lima puluh pixel. Jika garis bersentuhan maka garis tidak boleh bersentuhan lagi. Lingkaran harus memiliki diameter paling minimum tiga ratus pixel.



Gambar 2. Contoh lingkaran yang ingin dideteksi

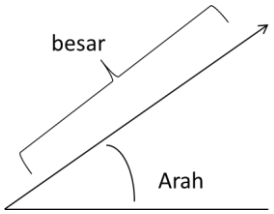


Gambar 3. Contoh bukan lingkaran

IV. TEORI VEKTOR

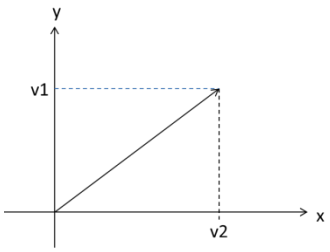
4.1 Definisi vektor

Vektor adalah kuantitas fisik yang memiliki besar dana arah.



Gambar 4. Vektor pada umumnya

Contoh Vektor pada R^2



Gambar 5. Vektor pada R2

Vektor Pada R2 dapat direpresentasikan dengan *tuple* dua buah konstanta saja. Berikut penulisan vektor R2

$$\vec{v} = (v1, v2)$$

Vektor nol adalah vektor yang semua komponennya adalah nol. Vektor nol dilambangkan dengan 0.

4.2 Operasi dasar pada vektor

Penjumlahan dua vektor bersifat asosiatif

$$\vec{v} + \vec{w} = \vec{w} + \vec{v}$$

Pengurangan dua vektor juga bersifat asosiatif

$$\vec{v} - \vec{w} = \vec{w} - \vec{v}$$

Perkalian vektor dengan skalar

$$k\vec{v} = (kv1, kv2, \dots, kvn)$$

Norma vektor adalah besar dari sebuah vektor. Norma vektor dapat dihitung menggunakan rumus berikut

$$|\vec{v}| = \sqrt{v1^2 + v2^2 + \cdots + vn^2}$$

4.2 Perkalian titik vektor

Terdapat 2 jenis perkalian vektor yaitu perkalian titik dan perkalian silang. Perkalian titik disebut juga perkalian dalam Euclid (Euclidean Inner product). Perkalian silang akan dibahas pada bagian selanjutnya.

Berikut cara melakukan perkalian titik

$$\vec{u}\,\vec{v} = |\vec{u}||\vec{v}|\cos(\propto)$$

atau

$$\vec{u}\,\vec{v} = u1v1 + u2v2 + \ \dots + unvn$$

Dengan alfa adalah sudut antara dua vektor.
Jika uv > 0 maka sudut alfa lancip
Jika uv = 0 maka sudut alfa tepat 90 derajat
Jika uv < 0 maka sudut alfa adalah sudut tumpul

4.3 Perkalian silang vektor

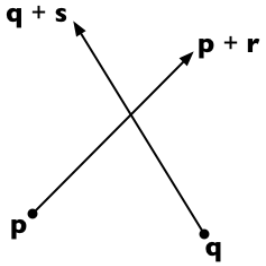
Perkalian silang berbeda dengan perkalian titik. Berikut rumus yang digunakan untuk melakukan perkalian silang.

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} =$$

$$= (a_2 b_3 - a_3 b_2)\mathbf{i} + (a_3 b_1 - a_1 b_3)\mathbf{j} + (a_1 b_2 - a_2 b_1)\mathbf{k}$$

4.4 Cara menentukan apakah dua vektor bersilangan

Terdapat sebuah cara yang mudah untuk menentukan apakah dua vektor bersilangan pada sebuah titik. Kedua vektor ini tentunya memiliki dua titik awal yang berbeda. Cara ini mencari titik potong dari dua vektor lalu mencari apakah kedua vektor mencapai titik potong tersebut.



Gambar 6. Illustrasi dua vektor bersilangan

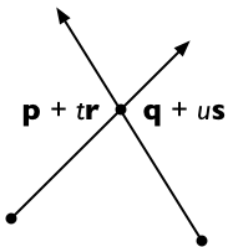
Metode ini membutuhkan operasi perkalian silang. Perkalian silang pada umumnya hanya dilakukan pada vektor tiga dimensi. Metode ini hanya membutuhkan besar dari hasil perkalian silang dua vektor ruang dua. Hasil perkalian dua vektor dua dimensi dapat dihitung dengan rumus berikut

$$\vec{v} \times \vec{w} = vx\, wy - vy\, wx$$

Pertama tentukan terlebih dahulu apakah ditemukan sebuah t dan u untuk rumus

p + t **r** = **q** + u **s**

pada gambar berikut



Gambar 7. Titik potong dua vektor

Penurunan rumus dari rumus awal memberikan sebuah cara untuk mendapatkan nilai t dan u. Cara mendapatkan nilai t dan u adalah dengan rumus berikut.

t = (**q** − **p**) × **s** / (**r** × **s**)

u = (**q** − **p**) × **r** / (**r** × **s**)

Dari dua rumus ini didapatkan nilai t dan u. Setelah didapatkan nilai t dan u maka dapat ditentukan apakah kedua vektor tersebut bersilangan.

Terdapat empat buah kasus

1. Kasus **r** × **s** = **0** dan (**q** − **p**) × **r** = **0**
   Pada kasus ini kedua vektor kolinear dan membutuhkan pemrosesan lebih lanjut mengunakan rumus berikut.
   t_0 = (**q** − **p**) · **r** / (**r** · **r**)
   t_1 = (**q** + **s** − **p**) · **r** / (**r** · **r**) = t_0 + **s** · **r** / (**r** · **r**)
   Jika range t_0 hinga t_1 bersentuhan dengan ranga angka 0 hingga 1 maka kedua vektor tersebut kolinear dan bersentuhan. Jika tidak maka kedua vektor tersebut kolinear dan tidak bersentuhan
2. Kasus **r** × **s** = **0** dan (**q** − **p**) × **r** ≠ **0**
   Pada kasus ini kedua buah vektor merupakan vektor parallel maka kedua vektor tidak bersentuhan
3. Kasus t dan u ditemukan dan $0 \le t \le 1$ dan $0 \le u \le 1$
   Pada kasus ini kedua vektor bersilangan pada titik **p** + t **r** = **q** + u **s.**
4. Kasus selain 3 kasus diatas
   Pada kasus ini kedua vektor tidak bersilangan [4]

V. Hipotesis

Berdasarkan definisi lingkaran pada bab tiga dibuat tiga algoritma yang diharapkan dapat mendeteksi lingkaran sesuai definisi tersebut.

Algoritma pendeteksi titik awal dan akhir mendekati
r = |vawal – vakhir|
Jika r < 150 maka titik awal dan akhir mendekati

Algoritma pendeteksi diameter minimum lingkaran
Untuk semua titik terhadap semua titik cari jarak maksimum antara kedua titik.
Simpanlah nilai maksimum tersebut dalam variabel d
Jika d > 300 maka diameter lingkaran minimum tercapai

Algoritma pendeteksi garis bersentuhan tidak lebih dari sekali
c = 0
p untuk setiap titik
  q untuk setiap titik yang ada sebelum p
    r = titik setelah p
    s = titik setelah q
    z = **r** × **s**
    y = (**q** − **p**) × **r**
    Jika z = 0 dan y = 0 maka
      t0 = (**q** − **p**) · **r** / (**r** · **r**)
      t1 = t_0 + **s** · **r** / (**r** · **r**)
      Jika 0 <= **t0** <= 1 atau 0 <= **t1** <= 1 maka
        c = c + 1

    t = (**q** − **p**) × **s** / (**r** × **s**)
    u = (**q** − **p**) × **r** / (**r** × **s**)
    Jika z ≠ 0 dan y ≠ 0 dan $0 \le t \le 1$ dan $0 \le u \le 1$ maka
      c = c + 1

Jika c ≤ 1 maka garis bersentuhan tidak lebih dari sekali

VI. METODE PENGAMBILAN DATA

Pada kesempatan kali ini berbagai macam data diambil menggunakan halaman web. Halaman web yang telah dilengkapi dengan penerima input berbasis *JavaScript.*

Javascript digunakan karena satu-satunya Bahasa pemrograman yang dapat digunakan pada halaman web adalah *Javascript*. Data yang didapatkan selanjutnya dikonversi secara manual kedalam format csv yang selanjutnya dapat diproses oleh berbagai macam perangkat lunak seperti *Microsoft Excel*.

VII. Hasil eksperimen percobaan pertama

Algoritma pertama memiliki tingkat akurasi yang tinggi yakni 94.73% dari tiga puluha delapan data terdapat dua yang tidak sesuai ekspektasi pengguna.

Algoritma kedua memiliki tingkat akurasi yang sangat rendah yakni 60%. Dari sepuluh data yang diambil empat data tidak sesuai ekspektasi pengguna.

Algoritma ketiga pada kasus tidak bersentuhan memiliki tingkat akurasi yang rendah yaitu 30%. Dari sepuluh data yang diambil tiga data sesuai ekspektasi pengguna.

Algoritma ketiga pada kasus bersentuhan satu kali memiliki tingkat akurasi yang sangat buruk yaitu 0%. Dari lima belas data yang diambil tidak satupun berhasil mendeteksi dengan benar.

Nilai minimum diameter yang menurut pengguna nyaman adalah dua ratus dua pixel.

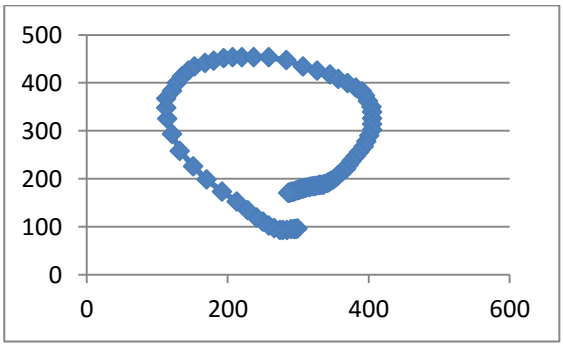
VIII. Analisis hasil percobaan pertama

Dari data yang didapatkan algoritma pertama sudah cukup baik bekerja pada tingkat akurasi 94.73%. Hipotesis terbukti untuk algoritma pertama.

Sedangkan algoritma kedua masih memiliki tingkat akurasi yang rendah. Algoritma kedua gagal karena kesalahan asumsi pada definisi lingkaran. Pada definisi lingkaran dikatakan bahwa sebuah lingkaran memiliki syarat berdiameter tiga ratus pixel. Namun, pada data yang didapatkan tiga ratus pixel merupakan angka yang terlalu besar. Nilai ini harus disesuaikan untuk meningkatkan tingkat akurasi.

Pada eksperimen berikutnya data akan menyesuaikan nilai diameter percobaan sebelunya. Nilai diameter yang disesuaikan adalah nilai diameter minimum yang menurut pengguna sesuai. Pengurangan nilai diameter minimum harus diikuti dengan kalibrasi jarak antara titik awal dan akhir yakni separuh dari diameter minimum.

Pada algoritma ketiga pada kasus tidak bersentuhan memiliki masalah karena batas akurasi perhitungan pada operasi matematika pada javascript. Titik yang terlalu dekat menghasilkan pembulatan yang akhirnya menyebabkan kesalahan pemeriksaan data. Berikut contoh kasus



Pada kasus ini pada awal dan akhir titik sangatlah berdekatan dan hasil perhitungan menghasilkan hasil yang salah. Solusi yang dilontarkan dari penulis adalah melakukan filtrasi pada data titik-titik tersebut. Filtrasi dilakukan dengan menghapus kedua titik yang terlalu dekat satu dengan yang lainnya. Terlalu dekat akan dikalibrasi hingga mendapatkan hasil yang baik.

Pada kasus ketiga kasus bersentuhan diasumsikan memiliki masalah yang sama dengan algoritma ketiga kasus tidak bersentuhan. Perlakuan yang sama akan diberikan pada kasus ini.

IX. Hasil Percobaan kedua

Hasil percobaan kedua menghasilkan hasil yang baik algoritma pertama berkurang akurasinya namun tidak banyak yakni 87% berkurang 7 persen saja dari percobaan pertama.

Hasil percobaan kedua untuk algoritma kedua menghasilkan 90% akurasi. Dari sepuluh data yang diambil Sembilan diantaranya sesuai dengan ekspektasi pengguna.

Hasil percobaan ketiga kasus tidak bersentuhan menghasilkan akurasi 100%. Dari sepuluh data yang diambil seluruhnya sesuai dengan ekspektasi pengguna.

Algoritma ketiga tidak bersentuhan dengan filtrasi data menghasilkan tingkat akurasi yang lebih tinggi yakni 100%. Dari sepuluh data yang diambil seluruhnya sesuai ekspektasi pengguna

Algoritma ketiga bersentuhan dengan filtrasi tetap memberikan akurasi yang buruk namun membaik yakni 40%. Dari sepuluh data yang diambil empat dinataranya memberikan hasil yang sesuai dengan ekspektasi pengguna.

V. Conclusion

Dari data yang didapatkan dan hasil analisa percobaan kali ini tidak berhasil membuat algoritma yang dapat dengan baik pada kasus tertentu.

REFERENCES

[1] http://www.gamasutra.com/view/news/237811/500_games_launched_per_day_on_iOS_last_year_and_other_digital_sales_facts.php
[2] http://www.gamedev.net/topic/601218-touch-screens-and-genre-barriers/
[3] https://uxmag.com/articles/gestures-animations-the-pillars-of-mobile-design
[4] http://stackoverflow.com/questions/563198/how-do-you-detect-where-two-line-segments-intersect
[5] http://dutchroll.sourceforge.net/dubsi/crossprod.html
[6] http://learningworksforkids.com/playbooks/fruit-ninja/
[7] http://www.automation-drive.com/EX/05-14-15/cnc3walls02.jpg

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, 8 Desember 2015

ttd

Candra Ramsi 13514090