

ANIMASI DAN VISUALISASI EKSPERIMEN BANDUL MATEMATIS DAN PEGAS DENGAN BAHASA PEMROGRAMAN TURBO PASCAL

Sri Hartati¹, Agus Harjoko¹ dan Agus Taufiq Mulyana²

¹Jurusan Fisika ²Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Gadjah Mada

Intisari

Telah dikembangkan perangkat lunak untuk animasi dan visualisasi eksperimen fisika dalam bidang mekanika untuk dua sistem bergerak yaitu Bandul Matematis dan Pegas dengan menggunakan bahasa pemrograman Turbo Pascal. Program animasi untuk sistem pegas tidak memerlukan nilai waktu karena ditujukan untuk menggambarkan peregangan pegas yang diberi beban. Teknik animasi yang digunakan yaitu teknik menggerakkan citra serta modifikasi dari teknik *inbetweening* dan teknik *sparse array*. Teknik-teknik tersebut dapat menghilangkan kedipan pada citra animasi. Hasil uji coba menunjukkan bahwa penggunaan teknik-teknik animasi yang disebutkan di atas menunjukkan dengan baik gerak harmonis dari bandul matematis menurut panjang tali dan sudut simpangan yang diberikan, dan gerak harmonis sistem pegas menurut beban yang diberikan. Namun, perangkat lunak ini masih menampilkan citra 2 dimensi. Secara keseluruhan animasi dapat berjalan dan terlihat cukup baik.

Kata kunci : Animasi, Visualisasi, Komputer grafika.

THE ANIMATION AND VISUALIZATION OF THE SIMPLE PENDULUM AND SPRING USING TURBO PASCAL LANGUAGE PROGRAMMING

Abstract

Software for animating simple pendulum and spring has been developed using Turbo Pascal. By relating between time and the motion of the simple pendulum the motion can be shown well. For the spring system animation, knowledge of time is not needed since the main purpose of the software is to show the relation between the length of the spring and the load applied to the spring. The animation is carried out by moving the image of the object being animated and by applying the modified *inbetweening* and *sparse array* techniques. These techniques eliminate blinking of the animated image. The result of experiment shows that the techniques mentioned above give a good result on performing simple harmonic motions of simple pendulum and spring system. However, the resulting software still displays two-dimensional images. In general the animation shows the motion well.

Keywords : Animation, Visualization, Computer graphics.

1. Pendahuluan

Ada banyak sistem dinamika dalam mekanika. Sistem bergerak tersebut dapat divisualisasikan dengan komputer. Program animasi dapat menjadi alat bantu dalam proses pengajaran fisika mekanika. Pengajaran teori akan lebih jelas jika disertai dengan visualisasinya.

Dalam memprogram suatu animasi sistem bergerak dalam mekanika, tidak hanya diperlukan pengetahuan tentang teknik animasi, tetapi juga perlu diketahui tentang sistem pewaktuan komputer. Ini dikarenakan sistem bergerak tersebut merupakan fungsi waktu. Cara menangani (menghitung) waktu juga memerlukan kejelian tersendiri supaya waktu yang terhitung sesuai dengan waktu yang berjalan. Sehingga animasi sistem bergerak tersebut dapat ditampilkan dengan baik.

Sistem bergerak yang akan dibahas di sini adalah bandul matematis yang mempunyai persamaan gerak fungsi waktu. Selain itu akan dibahas juga tentang animasi Pegas yang meregang karena diberi beban. Pada animasi pegas, waktu tidak menjadi masalah karena panjang akhir pegas tidak langsung mengikuti fungsi waktu.

II. Tinjauan Teori

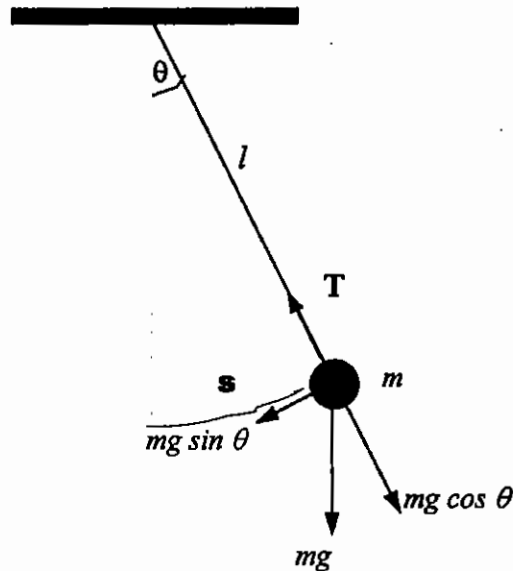
2. 1 Bandul matematis

Bandul matematis adalah sebuah titik massa yang digantungkan pada tali ringan yang tidak mulur kemudian diayunkan sehingga akan berosilasi. Gerak dari Bandul matematis ini mempunyai persamaan sebagai berikut (Serway dan Faughn, 1985)

$$s = l\theta_m \cos(\omega t + \phi) \quad (1)$$

Variabel-variabel pada persamaan (1) bersesuaian dengan Gambar 1. Sudut θ_m adalah sudut awal untuk θ . Nilai θ_m merupakan nilai maksimal untuk θ . Nilai ϕ tergantung pada posisi awal bandul saat disimpangkan. Variabel ω menunjukkan

kecepatan sudut osilasi yang besarnya sama dengan $\sqrt{g/l}$ dengan g adalah nilai percepatan gravitasi.



Gambar 1 Bandul Matematis dengan panjang tali l , gaya tegangan tali T , massa bandul m , percepatan gravitasi g , sudut simpangan θ dan panjang busur lintasan dari titik setimbang adalah s

2.2. Pegas

Salah satu contoh getran harmonis yang sederhana adalah bila suatu massa dikaitkan pada ujung pegas seperti ditunjukkan pada Gambar. 2. Jika pegas ditarik atau ditekan dari posisi setimbang dan dilepaskan maka akan timbul gaya sebesar

$$\vec{F} = -k\Delta x \quad (2)$$

Dengan k adalah tetapan gaya pegas dan Δx adalah nilai pertambahan atau pengurangan panjang pegas. Arah gaya berlawanan dengan arah pergeseran massa. Jika pegas ditarik sampai pada peregangan maksimum, A , dan dilepas maka energi potensial yang tersimpan $\frac{1}{2} kA^2$. Massa bergerak ke keadaan awal, sebagian energi tersebut diubah ke energi kinetik $\frac{1}{2} kv^2$, dan sehingga energi

potensial tersebut tereduksi menjadi $\frac{1}{2} kx^2$. Sehingga pada posisi terakhir diperoleh

$$v = \pm \sqrt{\frac{k}{m}(A^2 - x^2)} \quad (4)$$

Periode obyek dengan massa m diberikan oleh

$$T = 2\pi \sqrt{\frac{m}{k}} \quad (5)$$

dan frekwensi sistem pegas diberikan oleh

$$f = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (6)$$

Jika pegas tersebut digantung vertikal kemudian diberi beban maka akan ada gaya kearah atas sebesar \vec{F} dan ada gaya gravitasi kearah bawah sebesar $\vec{F}_g = m\vec{g}$, maka diperoleh persamaan $m\vec{g} = k\Delta x$ (7)

dengan m adalah nilai massa beban dan g adalah nilai percepatan gravitasi.

2.3 Teknik Animasi

Pada umumnya animasi dilakukan dalam suatu putaran/iterasi. Dalam setiap iterasi dilakukan pembaruan keadaan citra sehingga citra terkesan bergerak. Antara dua iterasi yang berurutan diberi selang waktu sesuai kecepatan animasi. Di sini akan dijelaskan 3 teknik animasi yaitu hapus-gambar (*inbetweening*), pergerakan citra dan larik jarang (*sparse array*).

2.3.1 Teknik animasi *inbetweening*

Teknik *inbetweening* pada prinsipnya adalah cara menghapus dan menggambar kembali. Mula-mula citra pada keadaan awal digambar. Selanjutnya, pada setiap iterasi dilakukan tiga hal yaitu menghapus citra pada iterasi sebelumnya, menentukan keadaan citra yang baru kemudian menggambar citra dalam keadaan yang baru tersebut (Hearn dan Baker, 1994).

2.3.2 Teknik animasi menggerakkan citra

Pada prinsipnya animasi dilakukan dengan menggambar obyek yang dianimasikan di suatu tempat lalu gambar tersebut dipindahkan ditempat lain setelah selang waktu tertentu. Gambar yang ditempatkan tersebut dapat dikenai operasi tertentu.

Teknik animasi menggerakkan citra ini didukung oleh prosedur *GetImage* dan *PutImage* pada Turbo Pascal (Yoyok, 1995). Prosedur *GetImage* untuk menyimpan citra yang ada pada suatu area segiempat pada layar ke dalam suatu variabel, deklarasinya adalah sebagai berikut

GetImage(x1, y1, x2, y2 : integer; var BitMap)

dengan (x1,y1) adalah titik kiri atas area yang ditempati citra, (x2,y2) adalah titik kanan bawah area yang ditempati citra serta BitMap adalah variabel tak bertipe tempat data-data citra disimpan. Prosedur *PutImage* untuk meletakkan citra yang telah disimpan oleh prosedur *GetImage* pada suatu tempat tertentu di layar, deklarasinya adalah sebagai berikut

PutImage(x, y : integer; var BitMap; BitBlit : word)

dengan (x,y) adalah titik kiri atas area yang akan ditempati citra, *BitMap* adalah variabel tak bertipe tempat data-data citra dan *BitBlit* adalah operasi biner yang akan dipakai untuk menampilkan citra pada layar. Ada 5 nilai yang mungkin untuk *BitBlit* yang disediakan Turbo Pascal sebagai konstanta. Yang terpenting di sini adalah *CopyPut* dan *XorPut*. Dengan menggunakan *CopyPut*, citra ditampilkan apa adanya. Dengan menggunakan *XorPut*, semua bit citra ditampilkan dengan cara dilakukan operasi biner *xor* dengan bit-bit pada layar pada posisi yang sama. Teknik ini cocok untuk citra statis. Cara penggunaan teknik ini dibedakan antara citra animasi yang mempunyai latar belakang dengan yang tidak mempunyai latar belakang atau warna latar belakangnya seragam. Program animasi yang dikembangkan di sini mempunyai warna latar belakang seragam, jadi di sini hanya akan dijelaskan animasi dengan warna latar belakang seragam.

Suatu citra yang sudah ditampilkan pada layar dapat dihapus dengan meletakkan citra itu kembali pada posisi yang sama dengan menggunakan prosedur *PutImage* dengan operasi biner *xor*. Ini bisa dilakukan karena suatu warna dioperasi-binarkan dengan warna itu sendiri akan menghasilkan warna latar belakang. Jadi citra pada iterasi sebelumnya dapat dihapus dengan operasi biner *xor*. Untuk menggambar citra pada keadaan sekarang digunakan *CopyPut*.

Pada cara di atas diperlukan dua kali pemanggilan prosedur *PutImage* pada suatu iterasi. Ada cara lain yang hanya memerlukan satu kali pemanggilan prosedur *PutImage*. Pemanggilan prosedur tersebut menggunakan operasi *CopyPut*. Cara ini dilakukan dengan memperhitungkan pergeseran maksimal baik terhadap sumbu x maupun terhadap sumbu y. Anggap pada keadaan awal citra terdapat pada area $(x1,y1,x2,y2)$ dan pergeseran maksimal terhadap sumbu x adalah *GeserX* serta pergeseran maksimal terhadap sumbu y adalah *GeserY* maka citra disimpan dengan instruksi

GetImage(x1-geserX,y1-geserY,x2+geserX,y2+geserY,citra);

Jika pada suatu iterasi, titik kiri atas citra yang baru adalah (x,y) maka citra diletakan dengan instruksi

PutImage(x-GeserX,y-GeserY,citra^,CopyPut);

2.3.3 Teknik *Sparse Array*

Animasi dengan teknik *sparse array* dicapai dengan menyimpan dulu semua keadaan citra animasi yang akan ditampilkan kemudian menampilkannya secara berurutan pada layar dengan selang waktu tertentu sehingga citra terkesan bergerak. Semua bit pada layar atau bagian tertentu pada layar diganti dengan bit-bit citra yang baru. Teknik ini pada prinsipnya sama dengan pembuatan film kartun (Hearn dan Barker, 1994).

2.4 Sistem pewaktuan komputer

Program animasi ini memanfaatkan *programmable interrupt timer* yang mengatur pewaktuan komputer yaitu pit 8253/8254 (terdapat 1 atau 2 buah).

Perangkat keras ini mempunyai 4 register utama yaitu counter 0, counter 1, counter 2 dan control register. Setiap counter mendapat masukan sinyal dengan frekwensi yang sama yaitu 1,19318 mhz, dan juga menghasilkan sinyal keluaran. Setiap counter dapat ditulisi atau dibaca dengan data sebesar 16 bit.

Counter 0 pada pit 8253/8254 ke-1 membentuk waktu sistem. Register ini mengeluarkan iterupsi perangkat keras irq0 yang dilayani rutin interupsi 08h (Messmer, 1997). Untuk membentuk waktu sistem ini, counter 0 beroperasi pada mode 2 yaitu *rate generator* (ada 6 mode yang mungkin). Selang waktu antara dua interupsi 08h yang berurutan tergantung dari nilai awal counter 0. Jika nilai awalnya n maka selang waktunya adalah

$$\Delta t = \frac{N}{1,19318} \mu s \quad (8)$$

Jika nilai awal adalah 0 maka n harus diisi 65536.

III. Implementasi

3.1 Animasi bandul matematis

Citra bandul matematis dibagi menjadi dua sub citra yaitu tali dan bola bandul. Tali yang berayun diimplementasikan dengan sebuah garis yang berubah-ubah sudutnya dengan titik pangkal yang tetap. Bola bandul diimplementasikan dengan lingkaran berisi.

3.1.1 Menentukan variabel bebas

Untuk kemudahan dan ketepatan waktu maka waktu dijadikan sebagai variabel bebas. Perangkat lunak hanya mengetahui nilai waktu dari perangkat keras sedangkan perangkat keras tidak setiap saat memperbarui nilai waktu. Jadi perangkat lunak tidak dapat menentukan nilai waktu dengan sembarang.

3.1.2 Menentukan nilai parameter untuk animasi bandul matematis

Untuk menentukan posisi titik pusat bola bandul pada layar digunakan rumus

$$x - x_0 = (l \sin \theta) \times skala \quad (9)$$

$$y - y_0 = (l \cos \theta) \times skala \quad (10)$$

dengan (x,y) adalah titik pusat bola bandul, (x_0,y_0) adalah titik pangkal tali yang tetap, $skala$ adalah banyaknya piksel yang mewakili 1 meter dan θ_t adalah sudut tali. Untuk mencari nilai θ_t digunakan rumus

$$\theta_t = \theta_m \cos(\omega t) \quad (11)$$

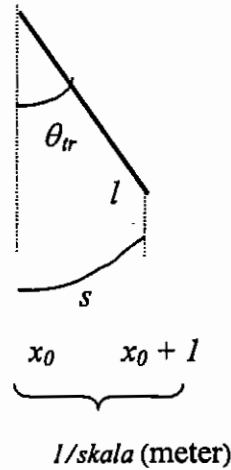
Selanjutnya dicari selang waktu antara dua iterasi yang berurutan (Δt) untuk memperbarui keadaan citra. Nilai Δt ini harus diatur sedemikian sehingga pergeseran citra baik terhadap sumbu x maupun sumbu y tidak terlalu besar. Pergeseran citra yang terbesar terjadi pada sekitar titik setimbang karena pada titik setimbang, Bandul Matematis mencapai kecepatan maksimal. Pada sekitar titik tersebut, pergeseran lebih besar terhadap sumbu x daripada terhadap sumbu y .

Pada titik setimbang nilai x sama dengan x_0 . Supaya pergeseran maksimal sebesar 1 piksel maka akan digunakan nilai pendekatan selang waktu antara $x = x_0 + 1$ dan $x = x_0$ untuk Δt (lihat Gambar 2). Untuk mencapai titik setimbang dari awal bola bandul dijatuhkan diperlukan waktu

$$t_s = \frac{\pi}{\omega} \quad (12)$$

Waktu yang diperlukan untuk mencapai titik dengan $x = x_0 + 1$ dari awal bola bandul dijatuhkan adalah

$$t_{i+1} = \arccos\left(\frac{1}{skala} \left(\frac{l\theta}{\omega}\right)\right) \quad (13)$$



Gambar 2 Lintasan antaran x_0 dan x_0+l sangat kecil sehingga $\sin \theta_t \cong \theta_t$

Disini dianggap sudut awalnya adalah positif. Dari persamaan (8) dan (9) dapat diketahui bahwa selang waktu antara titik dengan $x = x_0 + l$ dan $x = x_0$ adalah

$$\Delta t_s = t_s - t_{s+l} \quad (14)$$

Nilai Δt_s ini bukan nilai dari Δt yang dicari. Nilai Δt_s ini harus dikonversi ke suatu nilai yang bulat untuk nilai awal counter 0. Dari nilai awal counter 0 inilah dapat diketahui nilai Δt . Untuk keperluan ini, digunakan persamaan (8).

3.1.3 Teknik perhitungan waktu

Setelah mengisikan nilai awal yang disebutkan di atas maka nilai Δt dapat ditentukan dengan pasti. Setiap iterasi ditunggu terjadinya interupsi waktu (interupsi 08h) karena pada saat itu diketahui bahwa nilai waktu telah bertambah sebesar Δt . Setelah terjadi interupsi tersebut, proses perhitungan dan penggambaran citra kembali dimulai untuk memperbarui keadaan citra berdasarkan nilai waktu pada saat itu.

Karena di sini terjadi perubahan nilai awal counter 0 maka setelah animasi, nilai waktu pada alamat 0040h:006Ch perlu dikembalikan ke nilai sebenarnya jika nilai awal counter 0 adalah 0 (keadaan *default*). Untuk itu digunakan

hubungan

$$M_1 N_1 \cong M_2 N_2 \quad (15)$$

N_1 adalah nilai awal counter 0 selama proses animasi dan M_1 adalah jumlah interupsi 08h selama proses animasi. Selanjutnya M_2 adalah jumlah interupsi 08h yang seharusnya terjadi kalau nilai awal counter 0 adalah 0. Nilai N_2 harus diisi dengan nilai awal counter 0 dalam keadaan *default* yaitu 65536. Selanjutnya nilai M_2 ini ditambahkan ke nilai waktu sebelum proses animasi yang tentunya harus sudah disimpan.

3.1.4 Teknik animasi untuk gerakan tali bandul

Teknik yang digunakan adalah modifikasi dari teknik *inbetweening*. Garis (tali) yang tergambar sebelumnya tidak dihapus secara keseluruhan tetapi hanya titik-titik yang tidak ditempati lagi oleh citra tali yang baru. Untuk mengimplementasikan ini, garis tidak digambar dengan prosedur line yang disediakan oleh Turbo Pascal tetapi diproses titik per titik untuk memeriksa koordinatnya. Dalam hal ini garis dibuat dengan menggunakan algoritma Bresenham (Foley dan Van Dam, 1982). Untuk menampilkan citra tali yang sekarang dan menghapus citra tali sebelumnya diperlukan dua proses pembuatan garis. Yang satu untuk menghapus citra garis sebelumnya sedangkan yang satunya lagi untuk menggambar citra garis yang sekarang. Jika menggunakan prosedur line pada Turbo Pascal maka diperlukan dua kali pemanggilan prosedur tersebut. Begitu juga jika menggunakan algoritma Bresenham maka perlu dua blok iterasi untuk memproses dua garis tersebut, tetapi karena untuk memproses kedua garis tersebut dipilih variabel bebas yang sama yaitu y , dan kedua garis tersebut mempunyai titik pangkal yang sama dengan arah pergeseran nilai y yang sama yaitu membesar, maka dapat digunakan satu blok iterasi. Jadi di sini nilai koordinat y selalu sama untuk kedua garis pada setiap iterasi sedangkan nilai koordinat x bisa berbeda. Karena setiap iterasi nilai koordinat y pasti sama untuk kedua garis maka hanya perlu diperiksa nilai koordinat x -nya, jika berbeda maka titik pada garis sebelumnya dihapus dan titik pada garis sekarang digambar, jika sama maka tidak ada hal

yang perlu dilakukan.

3.1.5 Teknik animasi untuk gerakan bola bandul

Teknik yang digunakan adalah teknik animasi menggerakkan citra dengan tanpa menggunakan operasi biner *xor* untuk menghapus citra sebelumnya. Jadi diperlukan perluasan area segiempat yang berisi citra bola bandul yang digerakan. Karena telah diasumsikan bahwa pergeseran maksimal adalah sebesar 1 piksel baik terhadap sumbu *x* maupun sumbu *y* maka perluasannya sebesar 1 piksel ke kiri, kanan, atas dan bawah. Diperlukan perluasan keempat arah tersebut karena bola bandul dapat bergerak keempat arah tersebut. Jadi area segiempat yang disimpan akan berukuran dengan panjang dan lebar sama yaitu dua kali jari-jari bola bandul ditambah dua.

3.2 Animasi led indikator dan teknik perhitungan perioda

Led indikator ini digunakan sebagai penunjuk jika bola bandul berada pada sekitar titik setimbang. Led indikator ini akan menyala jika keadaan tersebut dicapai. Pada implementasinya dalam program, led indikator digambarkan dengan lingkaran berisi yang mempunyai warna berbeda antara menyala dengan tidak menyala. Led indikator dianggap menyala jika perbedaan antara absis titik pusat bola bandul dengan absis titik pangkal tali yang tetap lebih kecil atau sama dengan jejari bola bandul.

Jika N adalah cacah led menyala dan t adalah selang waktu antara pertama kali led menyala dengan terakhir kali led menyala maka perioda bandul matematis adalah

$$T = \frac{2t}{N} \quad (16)$$

Nilai perioda yang dihasilkan tidak tepat tetapi mengandung ralat. Saat-saat yang mungkin untuk awal perhitungan waktu dan akhir perhitungan waktu untuk nilai t pada persamaan (12) adalah pada saat led menyala. Saat-saat tersebut dipilih secara random dari iterasi-iterasi yang terjadi pada saat led

menyala. Jadi nilai waktunya merupakan nilai yang tercatat pada iterasi yang terpilih. Untuk keperluan ini digunakan pernyataan-pernyataan di bawah ini.

$$\begin{aligned} \text{Rand} &:= (\text{pi}/2 - \arccos((r_bandul/skala)/(l*abs(theta))))/w \\ P &:= 2*\text{trunc}(\text{rand}/dt) \end{aligned}$$

Pernyataan pertama akan mengakibatkan variabel *rand* berisi lamanya waktu antara saat sisi terluar bola bandul mencapai titik setimbang sampai titik pusat bola bandul mencapai titik setimbang. Lamanya waktu ini adalah setengah kali lamanya waktu led menyala. Logika penurunan pernyataan ini sama dengan persamaan (10), dengan variabel *w* bersesuaian dengan ω dan *theta* bersesuaian dengan θ_m . Selanjutnya pernyataan kedua menyebabkan variabel *p* berisi jumlah iterasi pada saat led menyala. Variabel *dt* bersesuaian dengan Δt .

Dengan diketahuinya jumlah iterasi pada saat led menyala dan iterasi awal led menyala maka proses pemilihan secara random iterasi-iterasi tersebut dapat dilakukan.

3.3 Teknik visualisasi pegas

Tujuan utama dari visualisasi percobaan pegas adalah tidak hanya untuk menunjukkan peregangan pegas karena diberi beban, tetapi disertai efek animasi yang membuat pegas benar-benar terlihat meregang.

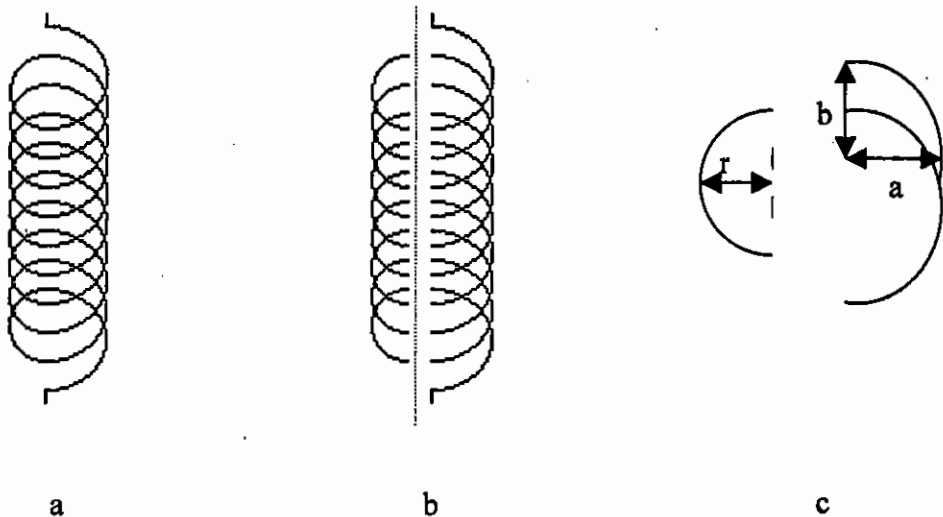
Citra pegas digambarkan dengan beberapa setengah ellips dan setengah lingkaran yang sambung menyambung seperti terlihat pada Gambar 3. Citra pegas tersebut dapat dibagi menjadi dua bagian. Bagian sebelah kiri terdiri dari beberapa setengah lingkaran dengan jari-jari *r*. Bagian sebelah kanan terdiri dari beberapa setengah ellips dengan jari-jari tegak *b* dan jari-jari mendatar *a*.

Jika terdapat *n* setengah ellips maka terdapat *n - 1* setengah lingkaran. Panjang citra pegas yang terdiri dari *n* setengah ellips adalah

$$L_p = 2nb - 2(n-1)r \quad (17)$$

Lebar citra pegas adalah

$$lb_p = r + a \quad (18)$$



Gambar 3 Visualisasi pegas

- a. Pegas digambarkan dengan beberapa buah setengah ellips dan setengah lingkaran
- b. Gambar pegas dapat dibagi dua bagian
- c. Komponen setengah ellips dan setengah lingkaran yang menyusun gambar pegas

3.4 Teknik animasi pegas

Teknik yang digunakan adalah modifikasi dari teknik *sparse array*. Perbedaannya dengan teknik *sparse array* yaitu pada teknik ini hanya digunakan satu *buffer* untuk menyimpan citra sedangkan pada teknik *sparse array* sebenarnya digunakan beberapa *buffer* untuk menyimpan citra-citra dalam keadaan yang berbeda yang kemudian ditampilkan secara berurutan. Pada teknik yang akan digunakan, *buffer* yang sudah berisi citra langsung ditampilkan pada layar kemudian isi *buffer* diubah sesuai keadaan citra selanjutnya lalu ditampilkan lagi. Karena di sini digunakan persamaan osilasi pegas vertikal yang merupakan fungsi waktu maka keadaan citra yang baru harus sesuai dengan waktu saat itu. Persamaan tersebut adalah

$$x = \Delta x \cos(\omega + \pi) + \Delta x \quad (19)$$

dengan Δx didapat dari persamaan (3) dan x adalah simpangan pegas terhadap titik setimbang.

Buffer yang digunakan adalah *array* dua dimensi yang cukup untuk menyimpan area segiempat yang ditempati titik-titik citra pegas pada layar. Karena hanya akan digunakan dua warna yaitu warna pegas dan warna latar pada area tersebut maka satu titik cukup diwakili satu bit. Bit-bit untuk titik yang bersesuaian akan diberi nilai 1 jika ditempati citra pegas dan 0 jika ditempati latar belakang. Selanjutnya citra ditampilkan dengan menggunakan prosedur *PutPixel* secara berurutan pada area segiempat yang telah ditentukan. Proses ini cukup lama sehingga jika ditampilkan proses osilasi dari pegas akan terlihat tidak sempurna, karena akan terjadi loncatan nilai panjang pegas yang disebabkan oleh rentang waktu yang lama antara proses penggambaran suatu keadaan citra dengan keadaan selanjutnya bahkan panjang pegas dapat memendek kembali walaupun masih jauh dari simpangan maksimal. Tetapi sampai di sini hal tersebut tidak terlalu menjadi masalah karena tujuan utama dari perangkat lunak yang akan dikembangkan hanya untuk menunjukkan peregangan pegas vertikal karena diberi beban.

IV. Kinerja Sistem

Program yang dikembangkan terdiri dari 5 bagian yaitu Unit GUI, Unit Bantu, Unit BandMat, Unit Pegas dan Program Mekanika sebagai program utama. Program ini pada pokoknya hanya berisi menu utama yang terdiri dari 4 item, yaitu

- **Bandul Matematis**: untuk menjalankan program animasi bandul matematis.
- **Pegas**: untuk menjalankan program animasi pegas.
- **Tentang Program**: untuk menampilkan identitas dari program mekanika.
- **Keluar**: untuk keluar dari program mekanika.

4.1 Program bandul matematis (Unit BandMat)

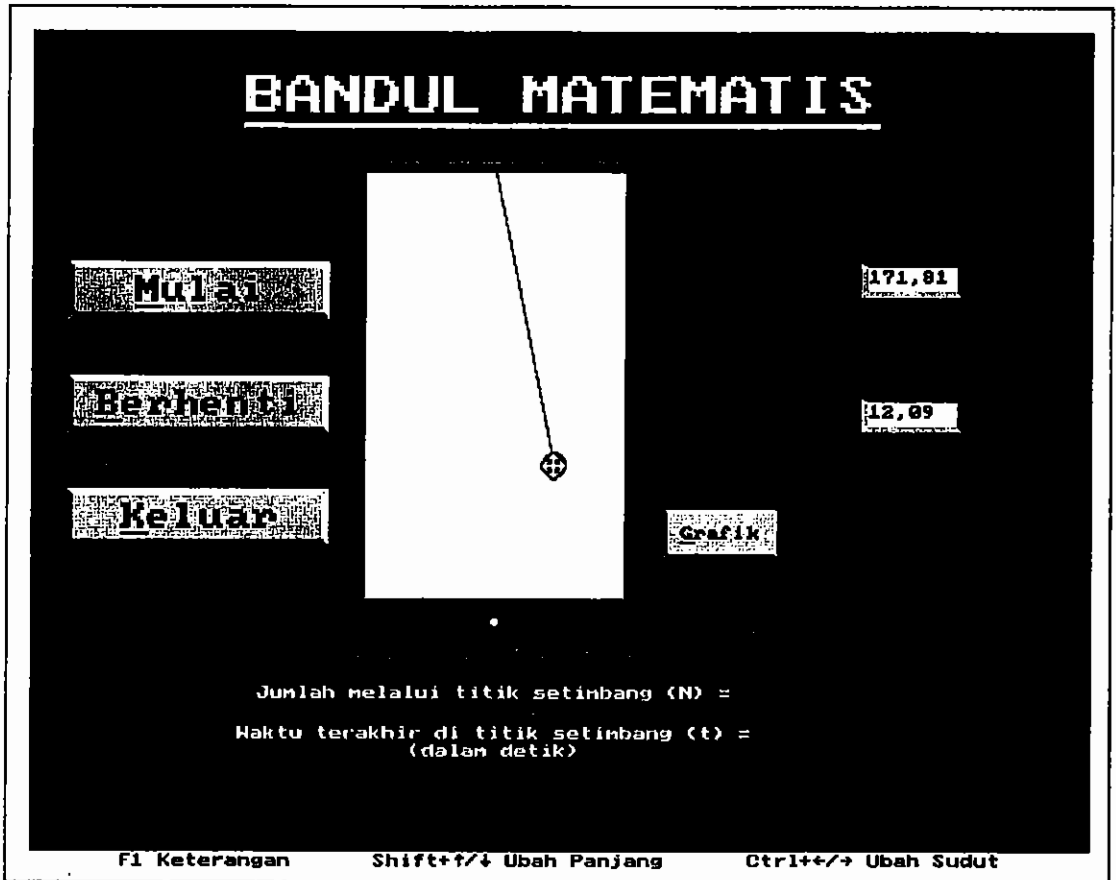
Antarmuka visualisasi eksperimen bandul matematis ditunjukkan pada Gambar 4. Unit program ini terdiri dari 2 kotak teks, 4 tombol dan beberapa menu. Penjelasanannya adalah sebagai berikut:

1. Kotak Teks

- “**Panjang Tali**” adalah tempat mengisi nilai panjang tali Bandul Matematis.
- “**Sudut Awal**” adalah tempat mengisi nilai sudut awal tali pada saat bola bandul dijatuhkan.

2. Tombol

- “**Mulai**” adalah untuk memulai percobaan ayunan (animasi Bandul Matematis) setelah diisi nilai panjang tali dan sudut awal.
- “**Berhenti**” adalah untuk menghentikan animasi ayunan setelah mulai.
- “**Keluar**” adalah untuk keluar dari program Bandul Matematis dan kembali ke menu utama (program Mekanika).
- “**Grafik**” adalah untuk menampilkan grafik hasil percobaan. Tombol ini akan muncul setelah dilakukan suatu percobaan. Setelah grafik ditampilkan, di bawah grafik terdapat tombol “**Tabel**”. Tombol ini untuk menampilkan tabel hasil percobaan.



Gambar 4 Tampilan program Bandul Matematis

3. Menu

- Antarmuka percobaan bandul matematis dilengkapi dengan menu bar *Keterangan*, *Ubah Panjang* dan *Ubah Sudut*. Kegunaan dari menu-menu tersebut adalah:

- *Keterangan*

Menu ini untuk menampilkan keterangan cara penggunaan program Bandul Matematis secara singkat. Untuk memilihnya maka tekan tombol keyboard **F1** atau klik tulisan '**F1**' pada bagian bawah layar.

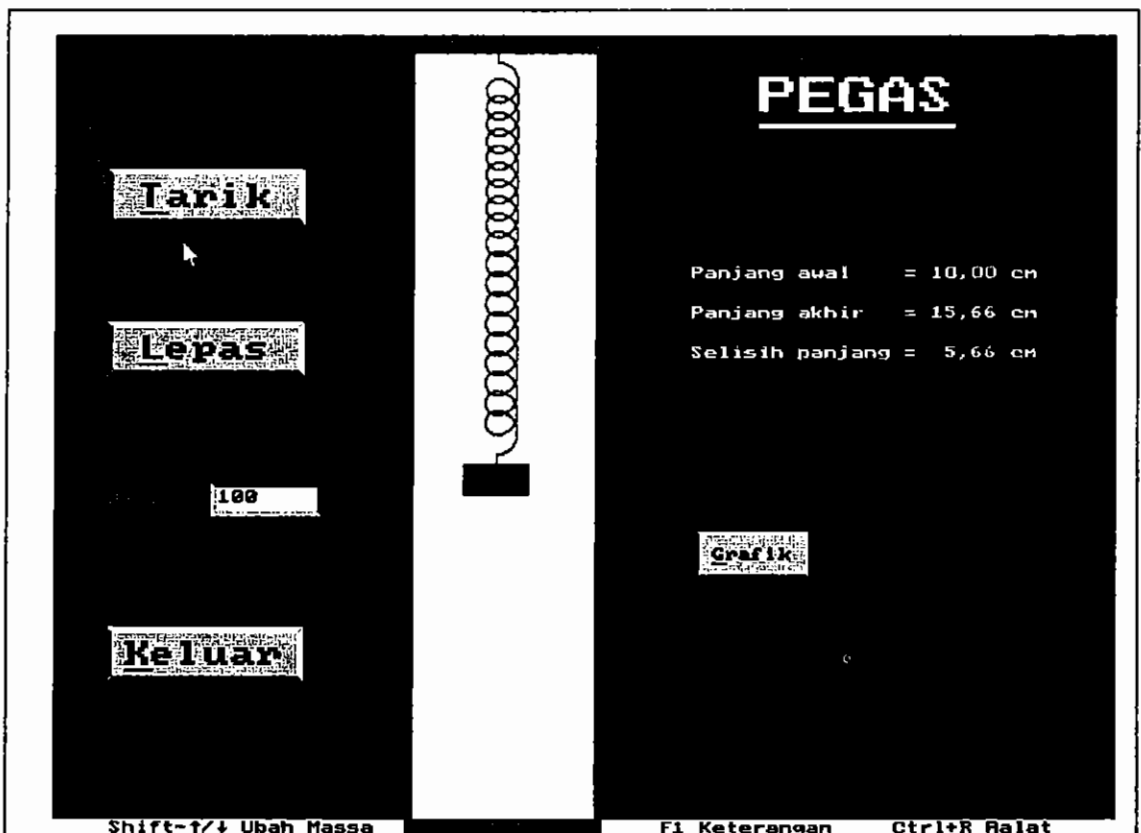
- *Ubah Panjang*

Untuk menurunkan nilai panjang tali maka tekan **Shift+↓** atau klik karakter '↓' pada bagian bawah layar dan untuk menaikannya maka tekan **Shift+↑** atau klik karakter '↑' pada bagian bawah layar. Aksi ini akan mengakibatkan nilai panjang tali selalu kelipatan 10.

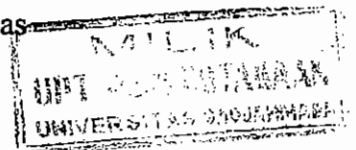
□ **Ubah Sudut**

Untuk menurunkan nilai sudut awal maka tekan **Ctrl+←** atau klik karakter '←' pada bagian bawah layar dan untuk menaikannya maka tekan **Ctrl+→** atau klik karakter '→' pada bagian bawah layar. Aksi ini akan mengakibatkan nilai sudut awal selalu bilangan bulat.

Selain menggunakan cara-cara di atas, juga bola bandul dapat di-*drag* ke posisi yang diinginkan. Nilai panjang tali dan sudut awal akan terisi sesuai dengan posisi bola bandul pada saat itu. Setelah tombol mouse dilepas maka bandul mulai berayun. Tetapi jika pada saat melepas tombol mouse tersebut ditekan **Shift** maka bandul akan diam. Untuk mengayunkannya maka klik tombol "**Mulai**".



Gambar 5 Tampilan program Pegas



4.2 Program Pegas (Unit Pegas)

Program ini terdiri dari 1 kotak teks, 4 tombol dan beberapa menu. Penjelasan nya adalah sebagai berikut.

1. Kotak Teks

- "Massa" adalah tempat mengisi nilai massa beban.

2. Tombol

- "Tarik" adalah untuk memulai percobaan yaitu menarik pegas dengan beban yang telah diisikan nilai massanya (memulai animasi Pegas).
- "Lepas" adalah untuk menghilangkan beban yang terlihat pada layar.
- "Keluar" adalah untuk keluar dari program Pegas dan kembali ke menu utama (program Mekanika).

3. Menu

- Antarmuka percobaan pegas dilengkapi dengan menu bar keterangan, *Ubah Massa* dan ralat. Kegunaan dari menu-menu tersebut adalah:

- *Ubah Massa*

Untuk menurunkan nilai massa beban maka tekan Shift+↓ atau klik karakter '↓' pada bagian bawah layar dan untuk menaikannya maka tekan Shift+↑ atau klik karakter '↑' pada bagian bawah layar. Aksi ini akan mengakibatkan nilai massa beban selalu kelipatan 10.

- *Keterangan*

Menu ini untuk menampilkan keterangan cara penggunaan program Pegas secara singkat. Untuk memilihnya maka tekan tombol keyboard F1 atau klik tulisan 'F1' pada bagian bawah layar.

- *Ralat*

Untuk memilih menu ini maka tekan Ctrl+R atau klik huruf 'R' pada bagian bawah layar yang berwarna merah. Menu ini untuk mengubah nilai ralat maksimal pada percobaan.

V. Penutup

Telah dikembangkan perangkat lunak untuk memvisualisasikan percobaan Bandul Matematis dan Pegas dengan menggunakan bahasa pemrograman Turbo Pascal. Perangkat lunak yang dikembangkan memperhitungkan nilai waktu yang sedang berjalan sehingga animasi dapat ditampilkan dengan baik. Pada program animasi ini nilai waktu hanya diperlukan oleh program animasi bandul matematis, sedang program animasi pegas tidak memerlukan waktu karena tujuan utamanya hanya untuk menggambarkan peregangan pegas yang diberi beban.

Teknik animasi yang digunakan yaitu teknik menggerakkan citra serta modifikasi dari teknik *inbetweening* dan teknik *sparse array*. Teknik-teknik yang digunakan tersebut dapat menghilangkan kedipan pada citra yang dianimasikan.

Hasil uji coba menunjukkan bahwa penggunaan teknik-teknik animasi yang disebutkan di atas menunjukkan dengan baik gerak harmonis dari bandul matematis menurut panjang tali dan sudut simpangan yang diberikan, dan gerak harmonis sistem pegas menurut beban yang diberikan. Namun, citra yang ditampilkan masih berupa citra 2 dimensi. Secara keseluruhan animasi dapat berjalan dan terlihat cukup baik. Perangkat lunak yang dihasilkan dalam penelitian ini dapat merupakan langkah awal untuk pengembangan visualisasi dan animasi eksperimen fisika dengan memperhitungkan parameter-parameter masukan dan keluaran yang dapat dikaitkan dengan besaran-besaran fisis eksperimen laboratorium.

Daftar Pustaka

- Foley, J.D., Van Dam, A., 1982, *Fundamentals of Interactive Computer Graphics*, *Addison-Wesley Publishing Company, Inc.*
- Hearn, D; Barker, P, 1994, *Computer Graphics*, 2nd edition, *Prentice Hall, New Jersey.*
- Messmer, H., 1997, *The Indispensable PC Hardware Book*, 3rd edition, *Addison-Wesley, New York*
- Serway,R.A., Faughn,J.S., 1985, *College Physics*, CBS College Publishing, New York.
- Yoyok, A.L., 1995, *Membuat Program Grafik Lewat Turbo Pascal 5.5/5.6*, edisi ke 2, Andi Offset, Yogyakarta