

## ANALISA RANGKAIAN ELEKTRONIKA DENGAN MENGGUNAKAN PROGRAM SIMULASI SPICE

Oleh : Lilik Eko Nuryanto

Staf Pengajar Jurusan Teknik Elektro – Politeknik Negeri Semarang

Jl.Prof. H. Soedarto, SH, Tembalang Semarang – 50275

E-mail: [lilikpolines@polines.ac.id](mailto:lilikpolines@polines.ac.id), [lilikpolines@gmail.com](mailto:lilikpolines@gmail.com)

### Abstrak

SPICE adalah singkatan dari *Simulation Program for Integrated Circuits Emphasis*. (Program simulasi untuk pengguna IC). PSpice merupakan versi PC dari SPICE (yang saat ini ada di OrCAD Corp. dari Cadence Design Systems, Inc). SPICE merupakan salah satu perangkat lunak yang baik untuk melakukan simulasi rangkaian elektronika baik analog maupun digital. SPICE merupakan sebuah program yang dapat melakukan simulasi rangkaian elektronika lewat komputer. Anda dapat menyaksikan bentuk gelombang dari tegangan maupun arus di rangkaian tersebut. SPICE ini menghitung tegangan serta arus versus waktu (Analisa Transien) atau juga versus frekuensi (Analisa AC). Selain itu tidak sedikit juga program ini digunakan untuk menganalisa sensitivitas, DC, derau dan distorsi. Suatu rangkaian elektronika dapat dengan lebih mudah dirancang dan dianalisa serta disimulasikan dengan bantuan program SPICE tersebut.

**Kata Kunci :** SPICE, PSPICE, OrCAD, NetList, Analisa, Simulasi, Rangkaian ,AC, DC,Transient

## 1. Pendahuluan

### 1.1. Tentang SPICE

SPICE (*Simulator Program with Integrated Circuit Emphasis*) adalah program komputer yang digunakan untuk menghitung (simulasi) perilaku rangkaian listrik. Program ini terutama diarahkan untuk perancangan rangkaian terpadu (IC). Data masukan program ini, yang disebut sebagai *input-deck* adalah berkas yang isinya merupakan paparan rangkaian yang hendak di-hitung-tiru. Ini meliputi tata hubungan (*topology*) rangkaian serta nilai-nilai parameter dari setiap elemen rangkaian, misalnya sumber arus (I), sumber tegangan (V), resistansi (R), kapasitansi (C), dan induktansi (L). Paparan rangkaian ini disebut sebagai *netlist*.

PSpice adalah versi PC dari SPICE (yang mana saat ini tersedia dari OrCAD Corp. Cadence Design Systems, Inc.). Versi mahasiswa (dengan kemampuan terbatas) dilengkapi dengan berbagai buku teks OrCAD edisi mahasiswa disebut PSpice AD Lite. Informasi tentang AD PSpice tersedia dari situs web OrCAD: <http://www.orcad.com/pspicead.aspx>.

Versi Light PSpice memiliki batasan berikut : rangkaian memiliki maksimum 64 node, 10 transistor dan 2 amplifier operasional.

SPICE bisa melakukan beberapa jenis analisa rangkaian, antara lain:

a. Analisa DC non-linier: menghitung kurva transfer DC.

Pada analisa DC, baik sumber tegangan maupun sumber arus yang digunakan mempunyai nilai yang tetap (konstan). Di dalam perancangan IC, analisa DC berguna antara lain untuk mengetahui titik operasi rangkaian dalam keadaan tanpa sinyal.

b. Analisa Transient dan Fourier non-linier: menghitung tegangan dan arus sebagai fungsi waktu ketika sinyal besar diberikan; Analisa Fourier memberikan frekuensi spektrum.

Pada analisa transient, sumber tegangan atau sumber arus mempunyai nilai yang merupakan fungsi dari atau berubah terhadap waktu. Bentuk fungsi waktu ini misalnya bisa berupa fungsi langkah

(*step-function*), atau berupa sebuah pulsa.

- c. Analisa AC Linier: menghitung output sebagai fungsi frekuensi. Sebuah plot dihasilkan.

Pada analisa AC, semua elemen rangkaian dianggap bersifat linier, sehingga analisa ini juga bisa disebut sebagai analisa sinyal kecil (*small-signal analysis*). Di sini sumber tegangan atau arus umumnya merupakan fungsi sinusoidal dengan nilai amplituda, frekuensi dan fase yang bisa dipilih. Selanjutnya dilakukan "penyapuan" (*sweeping*) dari nilai frekuensi sumber ini dari suatu nilai paling-rendah ke nilai paling-tinggi. Untuk setiap nilai frekuensi tersebut nilai besaran keluaran atau disebut pula sebagai "tanggapan" dari rangkaian, baik tegangan maupun arus pada terminal tertentu dihitung dan diperagakan.

- d. Analisa derau (*noise*)

Bagian perangkat program dari SPICE yang ini dapat digunakan untuk melakukan analisa terhadap derau yang dihasilkan oleh elemen rangkaian, misalnya resistor atau perangkat semikonduktor. Di sini, terminal input rangkaian dihubungkan dengan suatu sumber tegangan atau arus. Maka perangkat analisa ini akan melakukan perhitungan sumbangan derau dari setiap divais di dalam rangkaian tersebut terhadap terminal output yang diamati.

- e. Analisa parametrik  
f. Analisa *Monte Carlo*.

Sebagai tambahan, PSpice memiliki perpustakaan komponen standar analog dan digital seperti NAND, NOR, flip-flops, MUXes, FPGA, PLDs dan banyak lagi komponen digital. Ini membuatnya alat yang berguna untuk berbagai macam aplikasi analog dan digital.

Semua analisa bisa dilakukan pada suhu yang berbeda temperaturnya. Suhu standar adalah 300°K.

Rangkaian tersebut bisa berisi komponen berikut :

- a. Sumber tegangan dan arus independen dan dependen
- b. Resistor
- c. Kapasitor
- d. Induktor
- e. Mutual Induktor
- f. Line transmisi
- g. Penguat operasional
- h. Saklar
- i. Dioda
- j. Transistor Bipolar
- k. Transistor MOS
- l. JFET
- m. MOSFET
- n. Gerbang digital, dan
- o. komponen lainnya.

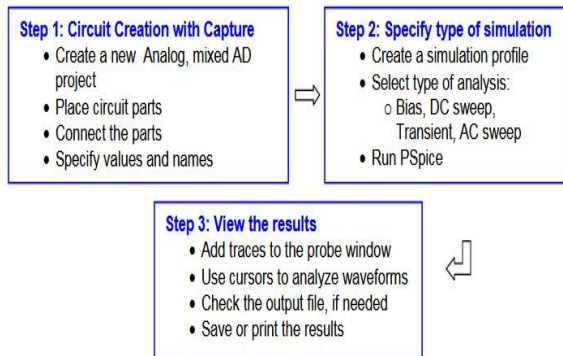
## 1.2. PSpice dengan OrCAD Capture (release 9.2 Lite Edition)

Sebelum seseorang dapat mensimulasikan sebuah rangkaian yang perlu dilakukan adalah menentukan konfigurasi rangkaian. Ini bisa dilakukan dengan berbagai cara. Salah satunya adalah dengan memasukkan deskripsi rangkaian sebagai file teks dalam bentuk elemen, koneksi, model elements dan jenis analisisnya. File ini disebut file input SPICE atau file sumber (*source file*). Cara alternatif adalah dengan menggunakan skema mencoba program seperti OrCAD CAPTURE. OrCAD Capture dibundel dengan PSpice Lite AD di CD yang sama yang disertakan dengan buku teks.

Capture adalah program yang *user-friendly* yang memungkinkan anda untuk menangkap skema dari rangkaian dan untuk menentukan jenis simulasi. Menangkap tidak hanya dimaksudkan untuk menghasilkan masukan untuk PSpice tapi juga untuk program desain layout PCD.

Urutan berikut merangkum tahapan yang berbeda yang terlibat dalam

mensimulasikan sebuah rangkaian dengan Capture dan PSpice:

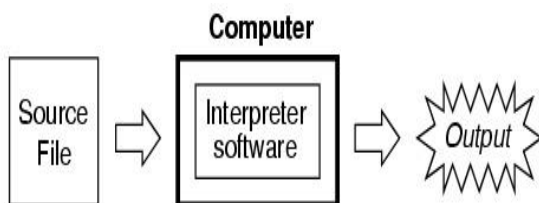


Gambar 1: Langkah-langkah yang terlibat dalam simulasi rangkaian dengan PSpice.

## 2. Dasar-Dasar Pemrograman SPICE

Pemrograman simulasi rangkaian elektronika dengan SPICE sangat mirip dengan pemrograman dalam bahasa komputer lain: anda harus menyetikkan perintah sebagai teks di file, simpan file itu ke *hard drive* komputer, kemudian proseskan isi file itu dengan sebuah program (kompilator atau interpreter) yang mengerti perintah tersebut.

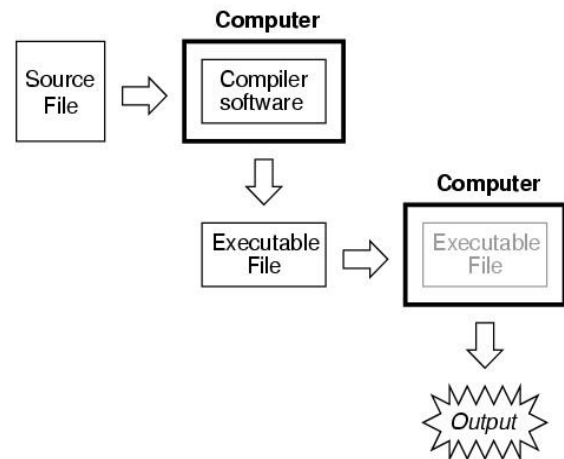
Dalam bahasa komputer yang diinterpretasikan, komputer menangani sebuah program khusus yang disebut *interpreter* yang menterjemahkan program yang anda tulis (yang disebut file sumber) ke dalam bahasa komputer sendiri, dengan cepat, saat dieksekusi:



Gambar 2: Bahasa *Interpreter*

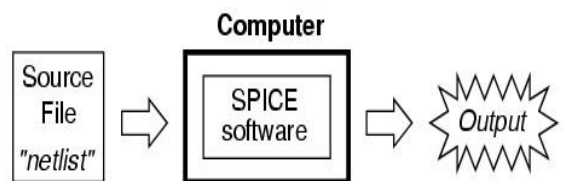
Dalam bahasa komputer yang dikompilasi, program yang anda tulis diterjemahkan sekaligus ke dalam bahasa komputer sendiri oleh sebuah program khusus yang disebut

*compiler*. Setelah program yang telah anda tulis telah "dikompilasi", file eksekusi yang dihasilkan tidak memerlukan terjemahan lebih lanjut untuk dipahami secara langsung oleh komputer. Sekarang bisa "dijalankan" di komputer apakah perangkat lunak *compiler* telah di-*instal* di komputer itu atau tidak.



Gambar 3: Bahasa *Compiler*

SPICE adalah bahasa yang diinterpretasikan (*interpreted language*). Agar komputer dapat memahami instruksi SPICE yang anda ketik, program komputer harus memiliki program SPICE (*interpreter*) yang terpasang.



Gambar 4: Blok Diagram Proses Interpretasi Program

File sumber SPICE sering disebut sebagai "*netlists*", meskipun kadang-kadang dikenal sebagai "*deck*" dengan setiap baris dalam file yang disebut "*kartu (card)*". *Netlists* diciptakan oleh orang seperti anda menyetikkan instruksi baris demi baris (*line-by-line*) menggunakan pengolah kata atau *editor* teks. *Editor* teks lebih disukai daripada pengolah kata untuk semua jenis pemrograman komputer, karena mereka

menghasilkan teks ASCII murni tanpa kode tertanam khusus untuk penyorotan teks (seperti huruf miring atau huruf tebal), yang tidak dapat diinterpretasikan oleh perangkat lunak *interpreter* dan *compiler*.

Seperti pada pemrograman umum, file sumber yang dibuat untuk SPICE harus mengikuti konvensi pemrograman tertentu. Ini adalah bahasa komputer itu sendiri, meski sederhana. Setelah diprogram di BASIC dan C / C ++, dan setelah beberapa pengalaman membaca program PASCAL dan FORTRAN, bahasa SPICE jauh lebih sederhana daripada semua ini. Ini tentang kompleksitas yang sama seperti bahasa *markup* misalnya HTML, kurang lebih seperti itu.

Ada langkah siklus yang harus diikuti dalam menggunakan SPICE untuk menganalisa sebuah rangkaian. Siklus dimulai saat pertama kali memanggil program pengeditan teks dan membuat draf pertama netlist anda. Langkah selanjutnya adalah menjalankan SPICE pada netlist baru dan lihat hasilnya. Jika Anda adalah pengguna pemula SPICE, usaha pertama anda untuk menciptakan netlist yang baik akan penuh dengan kesalahan sintaks yang kecil. Jangan khawatir seperti yang diketahui setiap programmer komputer, kemahiran hadir dengan banyak latihan. Jika percobaan anda menghasilkan pesan kesalahan atau hasil yang jelas tidak benar, anda perlu memanggil ulang program pengeditan teks dan memodifikasi netlist. Setelah memodifikasi netlist, anda perlu menjalankan SPICE lagi dan memeriksa hasilnya. Urutannya kemudian terlihat seperti ini:

- a. Buat netlist baru dengan program pengeditan teks. Simpan netlist tersebut ke file dengan nama pilihan anda.
- b. Jalankan SPICE pada netlist tersebut dan amati hasilnya.
- c. Jika hasilnya mengandung kesalahan, mulailah program pengeditan teks lagi dan ubah netlist.

- d. Jalankan SPICE lagi dan amati hasil baru.
- e. Jika masih ada kesalahan dalam output SPICE, edit ulang netlist lagi dengan program editing teks. Ulangi siklus edit / run ini sebanyak yang diperlukan sampai anda mendapatkan hasil yang diinginkan.
- f. Setelah anda "debugged" netlist anda dan mendapatkan hasil yang baik, jalankan SPICE lagi, hanya saja kali ini mengarahkan output ke file baru dan bukan hanya mengamati di layar komputer.
- g. Memulai program pengeditan teks atau program pengolah kata dan membuka file output SPICE yang baru saja anda buat. Ubah file yang sesuai dengan kebutuhan pemformatan Anda dan simpan perubahan tersebut ke disk dan / atau cetak di atas kertas.

Untuk menjalankan program "SPICE", anda perlu mengetikkan perintah pada antarmuka prompt terminal, seperti yang terdapat pada MS-DOS, UNIX, atau opsi MS-Windows DOS prompt:

```
spice < example.cir
```

Kata "*spice*" memanggil program interpretasi SPICE (dengan syarat perangkat lunak SPICE telah terinstal di komputer!), Simbol "<" mengalihkan isi file sumber ke interpreter SPICE, dan *example.cir* adalah nama dari file sumber untuk contoh rangkaian ini. Ekstensi file ".cir" tidak wajib; Saya telah melihat ".inp" (untuk "input") dan ekstensi ".txt" bekerja dengan baik juga. Bahkan akan bekerja bila file netlist tidak memiliki ekstensi. SPICE tidak peduli dengan nama file anda, asalkan memiliki nama yang kompatibel dengan file-system komputer anda (untuk mesin MS-DOS lama, misalnya, nama file harus panjangnya tidak lebih dari 8 karakter, dengan 3 ekstensi karakter, dan tidak ada spasi atau karakter non-alfanumerik lainnya).

Saat perintah ini diketik, SPICE akan membaca isi file *example.cir*, menganalisis rangkaian yang ditentukan oleh file tersebut, dan mengirim laporan teks ke output standar terminal komputer (biasanya layar, di mana anda dapat melihatnya). Sebuah output SPICE khas adalah beberapa layar layak informasi, jadi anda mungkin ingin melihatnya dengan sedikit modifikasi perintah:

```
spice < example.cir | more
```

Alternatif ini "menyalurkan" output teks SPICE ke utilitas "more", yang memungkinkan hanya satu halaman ditampilkan setiap kalinya. Apa artinya ini (dalam bahasa Inggris) adalah bahwa output teks SPICE dihentikan setelah satu layar penuh, dan menunggu sampai pengguna menekan tombol keyboard untuk menampilkan layar berikutnya yang penuh dengan teks. Jika anda hanya menguji contoh file rangkaian anda dan ingin memeriksa kesalahan apa pun, ini adalah cara yang baik untuk melakukannya.

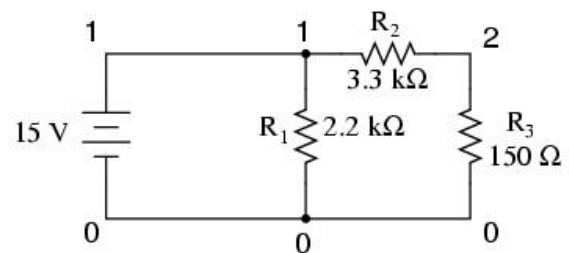
```
spice < example.cir > example.txt
```

Alternatif kedua ini (di atas) mengalihkan output teks SPICE ke file lain, disebut *example.txt*, di mana ia dapat dilihat atau dicetak. Pilihan ini sesuai dengan langkah terakhir dalam siklus pengembangan yang tercantum sebelumnya. Disarankan bahwa anda menggunakan teknik "redirection" ini ke file teks hanya setelah anda membuktikan contoh rangkaian netlist anda agar bekerja dengan baik, sehingga Anda tidak membuang waktu untuk meminta editor teks hanya untuk melihat hasilnya. selama tahap "debugging".

Setelah Anda memiliki output SPICE yang tersimpan dalam file *.txt*, Anda dapat menggunakan editor teks atau (lebih baik lagi!) Pengolah kata untuk mengedit hasilnya, menghapus spanduk dan pesan yang tidak perlu, bahkan menentukan font alternatif untuk menyorot judul dan / atau

data untuk tampilan yang lebih halus. Kemudian, tentu saja, anda bisa mencetak hasilnya ke kertas jika anda menginginkannya. Karena output SPICE langsung adalah teks ASCII biasa, file semacam itu akan dapat diinterpretasikan secara universal di komputer mana pun apakah SPICE terpasang di dalamnya atau tidak. Selain itu, format teks biasa memastikan bahwa file akan sangat kecil dibandingkan dengan file layar grafis yang dihasilkan oleh simulator "point-and-click".

Format file netlist yang dibutuhkan oleh SPICE cukup sederhana. File netlist tidak lebih dari sekadar file teks ASCII biasa yang berisi banyak baris teks, masing-masing garis menggambarkan komponen rangkaian atau perintah SPICE khusus. Arsitektur rangkaian ditentukan dengan menetapkan nomor ke titik koneksi masing-masing komponen di setiap baris, hubungan antara komponen yang ditunjuk oleh nomor umum. Periksa diagram rangkaian contoh berikut dan file SPICE yang sesuai. Harap diingat bahwa diagram rangkaian ada hanya untuk membuat simulasi lebih mudah dipahami oleh manusia. SPICE hanya mengerti *netlists*:



Gambar 5: Contoh rangkaian yang akan dibuat netlist-nya

Contoh netlist:

```
v1 1 0 dc 15
r1 1 0 2.2k
r2 1 2 3.3k
r3 2 0 150
.end
```

Setiap baris file sumber yang ditunjukkan di atas dijelaskan di sini:

- v1 mewakili baterai (sumber tegangan 1), terminal positif bernomor 1, terminal negatif bernomor 0, dengan keluaran tegangan DC 15 volt.
- R1 mewakili resistor R1 dalam diagram, dihubungkan antara titik 1 dan 0, dengan nilai 2,2 kΩ.
- R2 mewakili resistor R2 dalam diagram, dihubungkan antara titik 1 dan 2, dengan nilai 3,3 kΩ.
- R3 mewakili resistor R3 dalam diagram, dihubungkan antara titik 2 dan 0, dengan nilai 150 kΩ.

Poin umum secara elektrik (atau "nodus") dalam rangkaian rangkaian SPICE berbagi nomor yang sama, dengan cara yang sama seperti kabel yang menghubungkan titik umum di rangkaian besar biasanya memiliki label kawat biasa.

Untuk mensimulasikan rangkaian ini, pengguna akan mengetikkan enam baris teks pada editor teks dan menyimpannya sebagai file dengan nama unik (seperti example.cir). Setelah netlist disusun dan disimpan ke sebuah file, pengguna kemudian memproses file tersebut dengan salah satu pernyataan baris perintah yang ditunjukkan sebelumnya (spice <example.cir>), dan akan menerima output teks ini di layar komputer :

```

1*****10/10/17 ***** spice 2g.6 10/10/17
*****07:32:42*****

0example netlist

0**** input listing          temperature = 27.000 deg c

v1 1 0 dc 15
r1 1 0 2.2k
r2 1 2 3.3k
r3 2 0 150
.end

****10/10/17 ***** spice 2g.6 10/10/17
*****07:32:42*****

0example netlist
0**** small signal bias solution  temperature = 27.000 deg c
    
```

```

node voltage node voltage
( 1) 15.0000 ( 2) 0.6522

voltage source currents
name current

v1 -1.117E-02

total power dissipation 1.67E-01 watts

job concluded
0 total job time 0.02
1*****10/10/17 ***** spice 2g.6 10/10/17
*****07:32:42*****

0**** input listing          temperature = 27.000 deg c
    
```

SPICE dimulai dengan mencetak waktu, tanggal, dan versi yang digunakan di bagian atas output. Kemudian daftar parameter input (baris dari file sumber), diikuti oleh tampilan pembacaan tegangan DC dari masing-masing node (nomor referensi) ke ground (selalu nomor referensi 0). Hal ini diikuti oleh daftar pembacaan saat ini melalui setiap sumber tegangan (dalam kasus ini hanya ada satu, v1). Akhirnya, total disipasi daya dan waktu komputasi dalam hitungan detik dicetak.

Semua nilai output yang diberikan oleh SPICE ditampilkan dalam notasi ilmiah. Daftar keluaran SPICE yang ditunjukkan di atas adalah sedikit verbose untuk selera kebanyakan orang. Untuk presentasi terakhir, mungkin bagus untuk memangkas semua teks yang tidak perlu dan hanya menyisakan yang penting. Berikut adalah contoh output yang sama, yang diarahkan ke file teks (bumbu <example.cir> example.txt), kemudian dipangkas dengan bijaksana dengan editor teks untuk presentasi akhir dan dicetak:

```

Contoh netlist
v1 1 0 dc 15
r1 1 0 2.2k
r2 1 2 3.3k
r3 2 0 150
.end
    
```

```

node voltage node voltage
( 1) 15.0000 ( 2) 0.6522
voltage source currents
name current
v1 -1.117E-02
    
```

total power dissipation 1.67E-01 watts

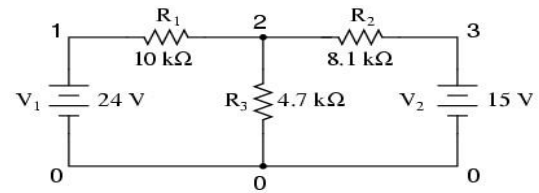
Salah satu hal yang sangat bagus tentang SPICE adalah format input dan output berupa teks biasa, yang merupakan format elektronik paling universal dan mudah diedit. Praktis komputer manapun akan bisa mengedit dan menampilkan format ini, meski program SPICE itu sendiri tidak dihuni komputer itu. Jika pengguna menginginkan, dia bebas menggunakan kemampuan canggih program pengolah kata untuk membuat keluaran terlihat lebih menarik. Komentar bahkan bisa disisipkan di antara lini output untuk kejelasan lebih lanjut kepada pembaca.

### 3. Rangkaian Elektronika dan Netlists

#### 3.1 Analisa Rangkaian Elektronika

Menggunakan Program Simulasi SPICE Rangkaian berikut adalah netlist yang telah teruji untuk SPICE 2g6, lengkap dengan deskripsi singkat. Tujuannya ada dua: untuk memberi contoh praktis tentang desain netlist SPICE untuk memahami lebih jauh tentang sintaks netlist SPICE, dan untuk menunjukkan bagaimana netlist sederhana dan kompak dapat digunakan dalam menganalisis rangkaian sederhana. Semua daftar keluaran untuk contoh-contoh ini telah "dipangkas" dari informasi asing, memberi Anda presentasi SPICE yang paling ringkas. Ini dilakukan terutama untuk menghemat ruang dalam dokumen ini. Keluaran khas SPICE berisi banyak header dan informasi ringkasan yang tidak selalu terkait dengan tugas yang ada. Jadi jangan kaget saat anda menjalankan simulasi sendiri dan temukan hasilnya sama sekali tidak seperti yang ditunjukkan di sini.

#### 3.2 Multiple-source DC resistor network, bagian 1



Tanpa kartu .dc dan kartu .print atau .plot, output untuk netlist ini hanya akan menampilkan tegangan untuk node 1, 2, dan 3 (dengan referensi ke node 0, tentu saja).

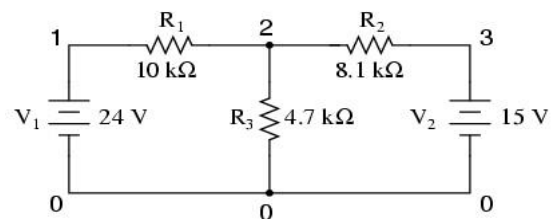
#### Netlist:

```
Multiple dc sources v1 1 0 dc 24 v2
3 0 dc 15 r1 1 2 10k r2 2 3 8.1k r3
2 0 4.7k .end
```

#### Output:

```
node voltage node voltage node
voltage ( 1) 24.0000 ( 2) 9.7470 (
3) 15.0000
voltage source currents name
current v1 -1.425E-03 v2 -6.485E-04
total power dissipation 4.39E-02
watts
```

#### 3.3 Multiple-source DC resistor network, bagian 2



Dengan menambahkan kartu analisis .dc dan menentukan sumber V1 dari 24 volt menjadi 24 volt dalam 1 langkah (dengan kata lain, tegangan 24 volt), kita dapat menggunakan analisis kartu .print untuk mencetak tegangan di antara dua titik yang kita inginkan. Anehnya, ketika opsi analisis .dc dipanggil, cetakan tegangan default untuk setiap simpul (ke ground) hilang, jadi kita akhirnya harus secara eksplisit menentukannya di kartu .print untuk melihat keduanya secara keseluruhan.

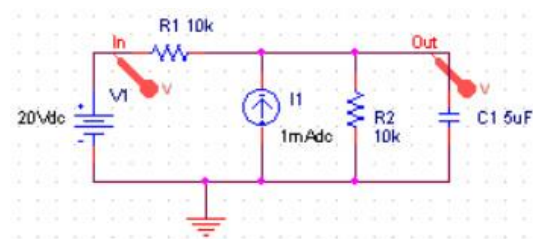
**Netlist:**

```
Multiple dc sources v1 1 0 v2 3 0
15 r1 1 2 10k r2 2 3 8.1k r3 2 0
4.7k .dc v1 24 24 1 .print dc v(1)
v(2) v(3) v(1,2) v(2,3) .end
```

**Output:**

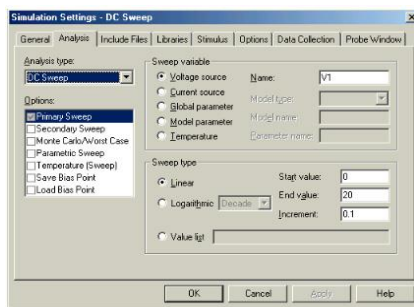
```
v1 v(1) v(2) v(3) v(1,2) v(2,3)
2.400E+01 2.400E+01 9.747E+00
1.500E+01 1.425E+01 -5.253E+00
```

**3.4 Simulasi DC Sweep**



Gambar 6: Rangkaian untuk DC Sweep

Kita akan menggunakan rangkaian pada Gambar atas yang akan dievaluasi Efeknya menyapu sumber tegangan (*DC Sweep*) antara 0 dan 20V. Kita akan menjaga sumber arus konstan pada 1mA. Pengesetannya seperti Gambar berikut:

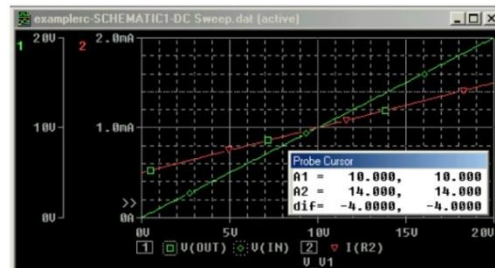


Gambar 7: Pengesetan untuk simulasi DC Sweep

Jalankan simulasi, PSpice akan menghasilkan file output yang berisi nilai semua tegangan dan arus dalam rangkaian tersebut. PSpice memiliki antarmuka yang *user-friendly* untuk menampilkan hasil

simulasi. Begitu simulasi selesai jendela Probe akan terbuka.

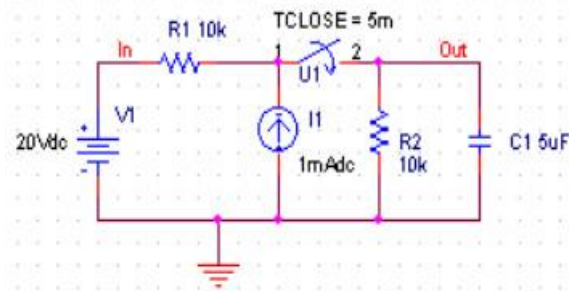
Hasil DC Sweep, menunjukkan  $V_{out}$ ,  $V_{in}$  dan arus yang lewat resistor R2. Kursor digunakan untuk  $V_{(out)}$  dan  $V_{(in)}$ . Hasil dari output PLOT nya seperti Gambar berikut:



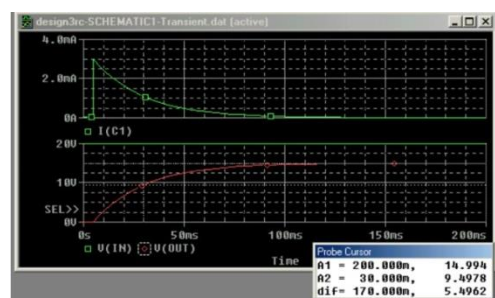
Gambar 8: Hasil DC Sweep, menunjukkan  $V_{out}$ ,  $V_{in}$  dan arus yang lewat resistor R2. Kursor digunakan untuk  $V_{(out)}$  dan  $V_{(in)}$ .

**3.5 Analisa Transient**

Kita akan menggunakan rangkaian yang sama seperti untuk DC Sweep, kecuali bahwa kita akan menerapkan tegangan dan sumber arus dengan menutup saklar, seperti yang ditunjukkan pada Gambar 9.



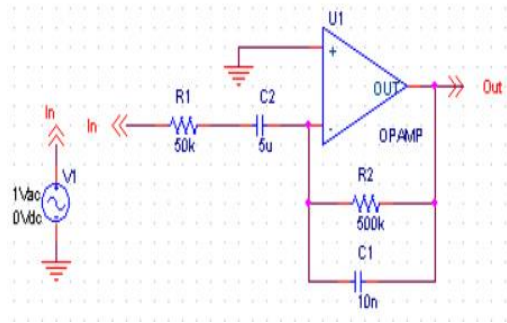
Gambar 9 : Rangkaian yang digunakan untuk simulasi transient



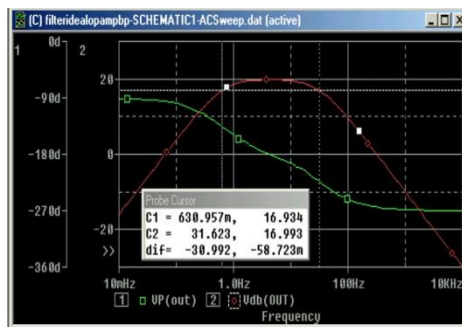
Gambar 10: Hasil simulasi transien pada Gambar 9.

### 3.6 AC Sweep pada Filter dengan Op-amp Ideal (Rangkaian Filter)

Rangkaian berikut akan disimulasikan dengan PSpice.

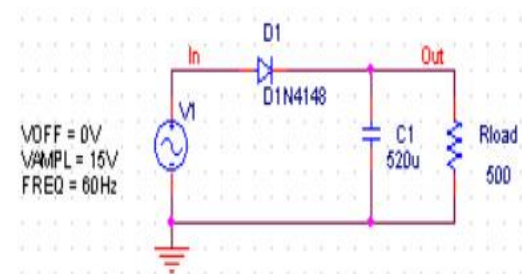


Gambar 11: Rangkaian Filter Aktif dengan Op-Amp Ideal.

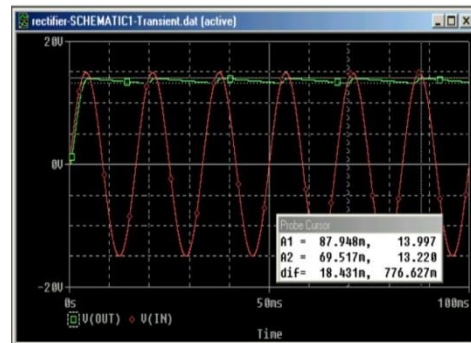


Gambar 12: Hasil AC Sweep dari Filter Filter Aktif pada gambar di atas.

### 3.7 Rangkaian Rectifier (peak detector) dan Penggunaan Parametric Sweep.



Gambar 13: Rangkaian Rectifier dengan Dioda D1N4148 dan Resistor Beban 500 Ohm.



Gambar 14: Hasil Simulasi Rangkaian Penyearah (Rectifier)

## 4. Penutup

SPICE merupakan sebuah simulator untuk rangkaian elektronika analog/digital yang sangat powerful yang dapat digunakan untuk meneliti dan memprediksi karakteristik rangkaian. Dirintis oleh Electronics Research Laboratory Universitas California, Berkeley (1975), nama SPICE adalah singkatan dari Simulation Program for Integrated Circuits Emphasis. PSpice adalah SPICE versi PC dari ORCAD Corp. of Cadence DesignSystems, Inc. Tersedia versi student dengan beberapa keterbatasan. Versi yang dipakai adalah PSpice Light 9.1 student version dengan keterbatasan, rangkaian maksimum mempunyai 64 titik, 10 transistor dan 2 op-amp. SPICE dapat mengerjakan beberapa analisa rangkaian, di antaranya yang penting adalah:

- Analisa *DC Non-linear*: menunjukkan kurva DC transfer
- Analisa *transient and Fourier Non-linear*: memperlihatkan kurva tegangan dan arus sebagai fungsi waktu
- Analisa *Fourier*: memperlihatkan spectrum frekuensi
- Analisa *AC Linear*: memperlihatkan grafik sebagai fungsi frekuensi
- Analisa *Derau (Noise)*
- Analisa *Parametric*

**DAFTAR PUSTAKA**

- Amit Kumar Singh, 2015, *Electronics Circuit SPICE Simulations with Ltspice*, CreateSpace Independent Publishing Platform, India
- A. Sedra and K. C. Smith, 2004, *Microelectronic Circuits with accompanying Rom CD containing Spice Circuit Examples*, Oxford University Press, England
- A. Vladimirescu, 1994, *The Spice Book*, J. Wiley & Sons, New York
- B. Carter, 2001, *Using Texas Instruments Spice Models in PSpice, Application Report*, Texas Instruments, Dallas
- Muhammad H. Rashid, 1995, *SPICE for Circuits and Electronics Using PSPICE (2nd Edition)*, Prentice Hall, New Jersey
- SPICE Tutorial,  
<http://www.seas.upenn.edu/~jan/spice/>