SIMULASI DENGAN R UNTUK AKTUARIS

Prana Ugiana Gio

Abstrak

Sebagai bagian dari profesi aktuaris, simulasi diperlukan untuk keperluan untuk menyelesaikan permasalahan yang tidak sederhana di bidang asuransi. Tulisan ini, merupakan sebuah pengantar atau tutorial terkait simulasi dalam R. Perangkat lunak R, dipilihnya perangkat lunak R, mengingat perangkat lunak ini bersifat open-source dan free.

Kata Kunci: Simulasi, tutorial, Asuransi

Simulasi dapat menjadi suatu teknik yang berguna untuk menyelesaikan permasalahan-permasalahan yang cukup rumit dalam asuransi umum (*general insurance*). Pada tulisan ini akan dipaparkan beberapa ide-ide dasar terkait simulasi, dan juga diberikan contoh-contoh melakukan simulasi secara manual dan dengan menggunakan komputer, yakni menggunakan bahasa pemrograman R.

1. Bilangan-Bilangan Acak dan Simulasi (*Random Numbers and Simulation*)

Permasalahan sering muncul pada analisis statistika (*statistical analysis*), yang mana memerlukan penggunaan bilangan-bilangan acak. Para peneliti (*survey samplers*) menggunakan bilangan-bilangan acak untuk memilih individu-individu acak dari suatu populasi dengan tujuan untuk mengestimasi parameter-parameter dari populasi. Para peneliti di bidang pertanian (*experimental agriculturalist*) sering kali menggunakan bilangan-bilangan acak, untuk menetapkan perlakuan-perlakuan (misalkan, perlakuan pemberian pupuk) terhadap blok-blok yang berbeda dengan cara acak (*random manner*), dengan maksud untuk menghindari

perbandingan bias antar perlakuan (*to avoid biasing comparisons between treatments*). Tugas semacam ini sangat bergantung pada bilangan-bilangan acak.

Para ahli teori (*the theoretician*) dapat menggunakan bilangan-bilangan acak untuk mempelajari distribusi dari suatu variabel acak yang rumit (*complicated random variable*), yang didefinisikan dalam variabel-variabel acak lainnya, yang mana distribusinya diketahui. Pengamatan-pengamatan acak (*random observations*) dari suatu variabel yang diketahui distribusinya, dibangkitkan/dihasilkan (*are generated*), dengan menggunakan bilangan-bilangan acak (*random numbers*), dan pengamatan-pengamatan acak ini kemudian digunakan untuk menghitung pengamatan-pengamatan acak dari variabel acak yang rumit. Dari nilai-nilai simulasi ini, para ahli teori (*theoretician*) dapat membuat kesimpulan mengenai distribusi yang tidak diketahui (bentuk (*shape*), rata-rata (*mean*), varians (*variance*), kemiringan (*skewness*), dan sebagainya). Berbagai tabel dari bilangan-bilangan acak telah dipublikasikan. Sebagai contoh:

- ⇒ *Fisher, R.A. and Yates, F. (1963). Statistical Tables for Use in Biological, Agricultural and Medical Research (sixth edition). Oliver and Boyd.*
- ⇒ *Rand Corporation. (1955). A Million Random Digits and 100 000 Normal Deviates. The Free Press, Glencoe, Illinois.*

Tabel 1 merupakan contoh dari tabel bilangan acak. Pada Tabel 1, tiap-tiap blok terdiri dari 5 baris. Tiap-tiap bilangan acak terdiri dari 5 digit. Tabel tersebut dipersiapkan oleh *Bureau of Transport Economics and Statistics of the Interstate Commerce Commission, Washington D.C.* Digit-digit individu (*individual digits*) atau pasangan-pasangan dari digit (*pairs of digits*) atau

kelompok-kelompok digit (*groups of digits*) dapat digunakan untuk tujuan pengambilan sampel (*sampling purposes*).

Tabel 1 Tabel Bilangan Acak

Line/Col.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
1	10 480	15 011	01 536	02 011	81 647	91 646	69 179	14 194	62 590	36 207	20 969	99 570	91 291	90 700
2	22 368	46 573	25 595	85 393	30 995	89 198	27 982	53 402	93 965	34 095	52 666	19 174	39 615	99 505
3	24 130	48 360	22 527	97 265	76 393	64 809	15 179	24 830	49 340	32 081	30 680	19 655	63 348	58 629
4	42 167	93 093	06 243	61 680	07 856	16 376	39 440	53 537	71 341	57 004	00 849	74 917	97 758	16 379
5	37 570	39 975	81 837	16 656	06 121	91 782	60 468	81 305	49 684	60 672	14 110	06 927	01 263	54 613
6	77 921	06 907	11 008	42 751	27 756	53 498	18 602	70 659	90 655	15 053	21 916	81 825	44 394	42 880
7	99 562	72 905	56 420	69 994	98 872	31 016	71 194	18 738	44 013	48 840	63 213	21 069	10 634	12 952
8	96 301	91 977	05 463	07 972	18 876	20 922	94 595	56 869	69 014	60 045	18 425	84 903	42 508	32 307
9	89 579	14 342	63 661	10 281	17 453	18 103	57 740	84 378	25 331	12 566	58 678	44 947	05 585	56 941
10	85 475	36 857	53 342	53 988	53 060	59 533	38 867	62 300	08 158	17 983	16 439	11 458	18 593	64 952
11	28 918	69 578	88 231	33 276	70 997	79 936	56 865	05 859	90 106	31 595	01 547	85 590	91 610	78 188
12	63 553	40 961	48 235	03 427	49 626	69 445	18 663	72 695	52 180	20 847	12 234	90 511	33 703	90 322
13	09 429	93 969	52 636	92 737	88 974	33 488	36 320	17 617	30 015	08 272	84 115	27 156	30 613	74 952
14	10 365	61 129	87 529	85 689	48 237	52 267	67 689	93 394	01 511	26 358	85 104	20 285	29 975	89 868
15	07 119	97 336	71 048	08 178	77 233	13 916	47 564	81 056	97 735	85 977	29 372	74 461	28 551	90 707
16	51 085	12 765	51 821	51 259	77 452	16 308	60 756	92 144	49 442	53 900	70 960	63 990	75 601	40 719
17	02 368	21 382	52 404	60 268	89 368	19 885	55 322	44 819	01 188	65 255	64 835	44 919	05 944	55 157
18	01 011	54 092	33 362	94 904	31 273	04 146	18 594	29 852	71 585	85 030	51 132	01 915	92 747	64 951
19	52 162	53 916	46 369	58 586	23 216	14 513	83 149	98 736	27 495	64 350	94 738	17 752	35 156	35 749
20	07 056	97 628	33 787	09 998	42 698	06 691	76 988	13 602	51 851	46 104	88 916	19 509	25 625	58 104
21	48 663	91 245	85 828	14 346	09 172	30 168	90 229	04 734	59 193	22 178	30 421	61 666	99 904	32 812
22	54 164	58 492	22 421	74 103	47 070	25 306	76 468	26 384	58 151	06 646	21 524	15 227	96 909	44 592
23	32 639	32 363	05 597	24 200	13 363	38 005	94 342	28 728	35 806	06 912	17 012	64 161	18 296	22 851
24	<u>29 334</u>	27 001	87 637	87 308	58 731	00 256	45 834	15 398	46 557	41 135	10 367	07 684	36 188	18 510
25	<u>02 488</u>	33 062	28 834	07 351	19 731	92 420	60 952	61 280	50 001	67 658	32 586	86 679	50 720	94 953
26	81 525	72 295	04 839	96 423	24 878	82 651	66 566	14 778	76 797	14 780	13 300	87 074	79 666	95 725
27	29 676	20 591	68 086	26 432	46 901	20 849	89 768	81 536	86 645	12 659	92 259	57 102	80 428	25 280
28	00 742	57 392	39 064	66 432	84 673	40 027	32 832	61 362	98 947	96 067	64 760	64 584	96 096	98 253
29	05 366	04 213	25 669	26 422	44 407	44 048	37 937	63 904	45 766	66 134	75 470	66 520	34 693	90 449
30	<u>91 921</u>	26 418	64 117	94 305	26 766	25 940	39 972	22 209	71 500	64 568	91 402	42 416	07 844	69 618

31	00582	04711	87917	77341	42206	35126	74087	99547	81817	42607	43808	76655	62028	76630
32	00725	69884	62797	56170	86324	88072	76222	36086	84637	93161	76038	65855	77919	88006
33	69011	65795	95876	55293	18988	27354	26575	08625	40801	59920	29841	80150	12777	48501
34	25976	57948	29888	88604	67917	48708	18912	82271	65424	69774	33611	54262	85963	03547
35	<u>09763</u>	83473	73577	12908	30883	18317	28290	35797	05998	41688	34952	37888	38917	88050
36	91567	42595	27958	30134	04024	86385	29880	99730	55536	84855	29080	09250	79656	73211
37	17955	56349	90999	49127	20044	59931	06115	20542	18059	02008	73708	83517	36103	42791
38	46503	18584	18845	49618	02304	51038	20655	58727	28168	15475	56942	53389	20562	87338
39	92157	89634	94824	78171	84610	82834	09922	25417	44137	48413	25555	21246	35509	20468
40	<u>14577</u>	62765	35605	81263	39667	47358	56873	56307	61607	49518	89656	20103	77490	18062
41	98427	07523	33362	64270	01638	92477	66969	98420	04880	45585	46565	04102	46880	45709
42	34914	63976	88720	82765	34476	17032	87589	40836	32427	70002	70663	88863	77775	69348
43	70060	28277	39475	46473	23219	53416	94970	25832	69975	94884	19661	72828	00102	66794
44	53976	54914	06990	67245	68350	82948	11398	42878	80287	88267	47363	46634	06541	97809
45	<u>76072</u>	29515	40980	07391	58745	25774	22987	80059	39911	96189	41151	14222	60697	59583
46	90725	52210	83974	29992	65831	38857	50490	83765	55657	14361	31720	57375	56228	41546
47	64364	67412	33339	31926	14883	24413	59744	92351	97473	89286	35931	04110	23726	51900
48	08962	<u>00358</u>	31662	25388	61642	34072	81249	35648	56891	69352	48373	45578	78547	81788
49	95012	68379	93526	70765	10592	04542	76463	54328	02349	17247	28865	14777	62730	92277
50	15664	10493	20492	38391	91132	21999	59516	81652	27195	48223	46751	22923	32261	85653
51	16408	81899	04153	53381	79401	21438	83035	92350	36693	31238	59649	91754	72772	02338
52	18629	81953	05520	91962	04739	13092	97662	24822	94720	06496	35090	04822	86774	98289
53	73115	35101	47498	87637	99016	71060	88824	71013	18735	20286	23153	72924	35165	43040
54	57491	16703	23167	49323	45021	33132	12544	41035	80780	45393	44812	12515	98931	91202
55	30405	83946	23792	14422	15059	45799	22716	19792	09983	74353	68668	30429	70735	25499
56	16631	35006	85900	98275	32388	52390	16815	69298	82732	38480	73817	32523	41961	44437
57	96773	20206	42559	78985	05300	22164	24369	54224	35083	19687	11052	91491	60383	19746
58	38935	64202	14349	82674	66523	44133	00697	35552	35970	19124	63318	29666	03387	59846
59	31624	76384	17403	53363	44167	64486	64758	75366	76554	31601	12614	33072	60332	92325
60	78919	19474	23632	27889	47914	02584	37680	20801	72152	39339	34806	08930	85001	87820

This table was prepared by the Bureau of Transport and Statistics of the Interstate Commerce Commission, Washington D.C. and is used herein with their permission.

This table was prepared by the Bureau of Transport and Statistics of the Interstate Commerce Commission, Washington D.C. and is used herein with their permission

Sebagai contoh, untuk memilih 30 orang secara acak dari suatu populasi yang terdiri dari 926 orang, kita dapat memberi nomor 926 orang tersebut, mulai dari 1 sampai 926 (setiap orang sudah diberi label angka, dari 1 sampai 926), dan kemudian pilih 30 bilangan-bilangan berdigit 3 (*and then choose 30 three-digit numbers*) dari tabel bilangan-bilangan acak. Jika suatu bilangan tiga digit (*if a three-digit number*) yang telah kita pilih tadi, yang mana bilangan tersebut berada di luar interval (*falls outside the range*) 1-926, kita abaikan (*discard*) bilangan tersebut, dan pilih kembali bilangan tiga digit selanjutnya (*and select three more digits*). Suatu

bilangan pecahan acak (*random fraction*) di antara 0 dan 1 dapat diperoleh dengan memilih suatu grup digit (*by choosing a group of digits*) (sebagai contoh 5 atau 10 digit) dan bayangkan titik desimalnya terletak di depan dari suatu grup digit tersebut (*and imagining a decimal point in front of the group*).

Tabel dari bilangan-bilangan acak harus digunakan secara berhati-hati, sedemikian rupa sehingga keacakan digit (*the randomness of the digits*) yang digunakan dipertahankan/dijaga dari satu kesempatan, ke kesempatan berikutnya (*in such a way the randomness of the digits used is preserved from one occasion to the next*). Salah satu cara adalah memulai pada awal tabel dan menyimpan berapa banyak tabel telah digunakan. Selanjutnya tabel (lain kali tabel) digunakan, memulai dengan nomor yang tidak digunakan pertama (*start with the first unused number*), dan lakukan melalui tabel secara bertahap (*and so work through the table gradually*). Pada contoh di depan, akan digunakan pendekatan ini.

Secara alternatif, buka tabel bilangan acak dengan keadaan mata terpejam (*without looking*), letakkan jari pada satu blok bilangan (*place a finger on a block of numbers*). Gunakan blok angka ini sebagai cara awal untuk memilih halaman, baris, kolom dan arah, kemudian angka-angka acak yang dibutuhkan dibaca (*use this block of numbers in a predetermined manner to choose the page, row, column and direction in which the required random numbers are to be read*).

Contoh 1 (Simulasi dengan Tabel Bilangan Acak)

Dalam suatu permainan peluang/kesempatan (*game of chance*), sebuah koin dilempar sebanyak 2 kali, dan jumlah kemunculan sisi kepala dicatat (*the number of 'heads' noted*). Selanjut koin tersebut dilempar lagi sebanyak 2 kali, dan jumlah kemunculan sisi ekor dicatat (*the number 'tails' in the latter two throws is noted*). Andaikan nilai maksimum dari jumlah kepala dan jumlah ekor merupakan X , yang mana jumlah dollar yang dibayar pemain ke bank.

Setiap digit-digit dalam tabel bilangan acak memiliki kemungkinan yang sama (*each of the digits in a table of random numbers is equally likely*). Jika digit tunggal (*single digit*) dipilih secara acak, maka ada 50% peluang/kesempatan (*chance*) bahwa digit akan bernilai kurang dari atau sama dengan 4 (0,1,2,3,4) dan 50% peluang/kesempatan (*chance*) digit akan bernilai lebih besar dari 4 (5,6,7,8,9). Ini memungkinkan untuk melakukan simulasi permainan dengan memilih digit-digit dari tabel bilangan acak. Jika suatu digit bernilai kurang dari atau sama dengan 4, kita akan mencatat 'kepala', jika tidak 'ekor'. Digit-digit acak untuk simulasi ini diringkas pada Tabel 2 yang mana diambil dari kolom pertama dari Tabel 1.

Tabel 2 Simulasi Permainan Kesempatan (*Simulation of a Game of Chance*)

Simulation number	First random digit	Second random digit	Number of 'heads'	Third random digit	Fourth random digit	Number of 'tails'	Payment (\$)
1	1	0	2	4	8	1	2
2	0	2	2	2	3	0	2
3	6	8	0	2	4	0	0
4	1	3	2	0	4	0	2
5	2	1	2	6	7	2	2
6	3	7	1	5	7	2	2
7	0	7	1	7	9	2	2
8	2	1	2	9	9	2	2
9	5	6	0	2	9	1	1
10	6	3	1	0	1	0	1
11	8	9	0	5	7	2	2
12	9	8	0	5	4	1	1
13	7	5	0	2	8	1	1
14	9	1	1	8	6	2	2
15	3	5	1	5	3	1	1
16	0	9	1	4	2	0	1
17	9	1	1	0	3	0	1
18	6	5	0	0	7	1	1
19	1	1	2	9	5	2	2
20	1	0	2	8	5	2	2
21	0	2	2	3	6	1	2
22	8	0	1	1	0	0	1
23	1	1	2	5	2	1	2
24	1	6	1	2	0	0	1
25	7	0	1	5	6	2	2
26	4	8	1	6	6	2	2
27	3	5	1	4	1	0	1
28	6	4	1	3	2	0	1
29	6	3	1	9	2	1	1
30	9	3	1	3	4	0	1
Total							44

Pada Tabel 2, diketahui:

⇒ Untuk simulasi ke-1, diketahui:

- nilai *first random digit* adalah 1. Nilai $1 \leq 4$, maka 'kepala'.
- nilai *second random digit* adalah 0. Nilai $0 \leq 4$, maka 'kepala'.
 - sehingga jumlah kepala adalah 2 (*number of 'heads'*).
- nilai *third random digit* adalah 4. Nilai $4 \leq 4$, maka 'kepala'.
- nilai *fourth random digit* adalah 8. Nilai $8 > 4$, maka 'ekor'.
 - sehingga jumlah ekor adalah 1 (*number of 'tails'*).

⇒ Maka $payment (\$) = \max(jumlah\ kepala = 2, jumlah\ ekor = 1) = 2$.

Diketahui bilangan-bilangan acak pada baris pertama dan kolom pertama adalah 10 480. Dari 5 bilangan tersebut, diambil 4 bilangan, yakni:

⇒ 1 untuk *first random digit*

⇒ 0 untuk *second random digit*

⇒ 4 untuk *third random digit*

⇒ 8 untuk *fourth random digit*

⇒ Untuk simulasi ke-2, diketahui:

- nilai *first random digit* adalah 0. Nilai $0 \leq 4$, maka 'kepala'.
- nilai *second random digit* adalah 2. Nilai $2 \leq 4$, maka 'kepala'.
 - sehingga jumlah kepala adalah 2.
- nilai *third random digit* adalah 2. Nilai $2 \leq 4$, maka 'kepala'.
- nilai *fourth random digit* adalah 3. Nilai $3 \leq 4$, maka 'kepala'.
 - sehingga jumlah ekor adalah 0.

⇒ Maka $payment (\$) = \max(jumlah\ kepala = 2, jumlah\ ekor = 0) = 2$.

Diketahui bilangan-bilangan acak pada baris pertama dan kolom pertama adalah 10 480.

Bilangan-bilangan acak pada baris kedua dan kolom pertama adalah 22 368.

⇒ 0 untuk *first random digit* (angka 0 ini merupakan angka terakhir pada bilangan acak baris pertama dan kolom pertama)

- ⇒ 2 untuk *second random digit* (angka ini merupakan angka pertama pada bilangan acak baris kedua dan kolom pertama)
- ⇒ 2 untuk *third random digit* (angka ini merupakan angka kedua pada bilangan acak baris kedua dan kolom pertama)
- ⇒ 3 untuk *fourth random digit* (angka ini merupakan angka ketiga pada bilangan acak baris kedua dan kolom pertama)

Dari simulasi ini, disimpulkan bahwa peluang untuk membayar \$0 adalah sangat kecil dan peluang untuk membayar \$1 atau \$2 hampir sama (*almost equal*). Estimasi dari probabilitas-probabilitas tersebut adalah

$$P(\text{membayar } \$0) = \frac{1}{30} = 0,03;$$

$$P(\text{membayar } \$1) = \frac{14}{30} = 0,47;$$

$$P(\text{membayar } \$2) = \frac{15}{30} = 0,50;$$

Rata-rata pembayaran (*average payment*) = $\frac{44}{30} = \$1,47$. Estimasi yang lebih tepat (*more precise estimates*) dari probabilitas dan rata-rata pembayaran dapat diperoleh dengan meningkatkan jumlah simulasi.

Further reading: Pollard [18] 236-7; Pooch [19] 1-32; Stuart & Ord [23] 324-9

Contoh 2 (Simulasi dengan R)

Suatu koin misalkan akan dilempar sebanyak 1000 kali. Secara teori, peluang untuk muncul sisi kepala (*head*) adalah 0,5, dan juga peluang untuk muncul sisi ekor adalah 0,5 (*tail*). Jika koin tersebut dilempar sebanyak 1000 kali, secara teori muncul sisi kepala sebanyak 500 dan muncul sisi ekor sebanyak 500 juga. Namun dalam prakteknya, apakah akan memperoleh hasil demikian? Gambar 1 merupakan contoh simulasi untuk percobaan melempar sebuah koin sebanyak 1000 kali. Misalkan angka 1 menyatakan kejadian muncul kepala dan angka 0 menyatakan kejadian muncul ekor. Pada Gambar 2, perintah R

rbinom (n=1000, size = 1, prob=.5)

dapat diartikan melakukan percobaan sebanyak $n = 1000$ kali, dengan probabilitas untuk terjadi sukses sebesar 0,5. Misalkan kejadian sukses adalah kejadian muncul sisi kepala. Perhatikan bahwa $size = 1$ berarti dalam suatu percobaan hanya memiliki 2 kejadian yang mungkin, yakni 0 dan 1. Berdasarkan Gambar 1:

- ⇒ Untuk percobaan pertama muncul sisi ekor (angka 0).
- ⇒ Untuk percobaan kedua muncul sisi ekor (angka 0).
- ⇒ Untuk percobaan ketiga muncul sisi kepala (angka 1), dan seterusnya.

```

1 percobaan = rbinom(n=1000, size=1, prob=.5) #Melakukan percobaan melempar koin sebanyak 1000 kali
2 percobaan
3 percobaan = table(percobaan)
4 print(percobaan)
5

1:1 (Top Level) R S

Console ~/
> percobaan = rbinom(n=1000, size=1, prob=.5) #Melakukan percobaan melempar koin sebanyak 1000 kali
> percobaan
[1] 0 0 1 1 1 0 0 0 0 1 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0
[52] 1 1 1 0 0 1 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1
[103] 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1
[154] 0 1 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 1 1 0 0 0 0 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 1 1 1 1
[205] 1 0 0 1 0 1 0 0 1 1 1 0 1 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1
[256] 1 0 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1
[307] 1 1 1 1 1 1 0 1 0 1 0 1 1 0 1 1 1 1 1 0 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 1 0 0 0 0
[358] 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1 1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1
[409] 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 1 1 1 1 0
[460] 0 0 0 1 1 0 0 0 1 0 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1
[511] 0 1 1 0 0 0 1 0 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 1 0 1 0 0 0 0 1 1 0 1 1 1 0 0 0 1 1 0 1 1 0 1 0 1 1 0 0
[562] 0 0 0 0 0 1 1 1 0 1 1 0 1 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1
[613] 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 0 0 1 0 0 1
[664] 1 0 0 1 1 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 1 1 0 1 1 1 0 1
[715] 1 0 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 1 1 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1
[766] 1 1 1 0 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 0 0
[817] 0 1 0 0 1 0 0 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 1 1 0 0 1 0 0 1
[868] 1 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0 1 0 0 1 1 1 1 1 1 0
[919] 0 1 1 1 1 0 0 1 1 0 0 0 1 1 1 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 1 1 1 0 1 1 1 1
[970] 1 1 1 0 0 0 1 1 0 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 0 0

> percobaan = table(percobaan)
> print(percobaan)
percobaan
 0    1
484 516

```

Gambar 1 Simulasi Percobaan Melempar Sebuah Koin sebanyak 1000 Kali dengan R

Dari 1000 percobaan, diketahui muncul sisi kepala (angka 1) sebanyak 516, sehingga estimasi probabilitas sebenarnya untuk sisi kepala adalah $\frac{516}{1000} = 0,516$. Berikut disajikan kode R pada Gambar 1.

```

percobaan = rbinom(n=1000, size=1, prob=.5) #Melakukan percobaan melempar koin sebanyak
1000 kali
percobaan
percobaan = table(percobaan)
print(percobaan)

```

Contoh 3 (Simulasi dengan R)

Kode program R berikut untuk mengilustrasikan Contoh 1.

```
bilangan_acak_pertama = floor((10 * runif(n=30) ) )
bilangan_acak_kedua = floor((10 * runif(n=30) ) )
bilangan_acak_ketiga = floor((10 * runif(n=30) ) )
bilangan_acak_keempat = floor((10 * runif(n=30) ) )

head_1 = ifelse(bilangan_acak_pertama <= 4,1,0)
head_2 = ifelse(bilangan_acak_kedua <= 4,1,0)
tail_1 = ifelse(bilangan_acak_ketiga > 4,1,0)
tail_2 = ifelse(bilangan_acak_keempat > 4,1,0)

jumlah_kepala = head_1 + head_2
jumlah_ekor = tail_1 + tail_2

for(i in 1 : length(bilangan_acak_pertama))
{
jumlah_yang_dibayar_dollar[i] = max(jumlah_kepala[i], jumlah_ekor[i])
}

tabel = data.frame(bilangan_acak_pertama, bilangan_acak_kedua, jumlah_kepala,
bilangan_acak_ketiga, bilangan_acak_keempat, jumlah_ekor, jumlah_yang_dibayar_dollar)

print(tabel)

print(table(tabel[, "jumlah_yang_dibayar_dollar"]) / length(bilangan_acak_pertama))
```

Gambar 2

```

> print(tabel)
  bilangan_acak_pertama bilangan_acak_kedua jumlah_kepala bilangan_acak_ketiga bilangan_acak_keempat jumlah_ekor jumlah_yang_dibayar_dollar
1                3                2                2                1                4                0                2
2                7                7                0                1                3                0                0
3                6                6                0                6                1                1                1
4                0                8                1                4                6                1                1
5                9                2                1                6                7                2                2
6                6                0                1                6                3                1                1
7                4                2                2                0                2                0                2
8                4                8                1                1                2                0                1
9                8                6                0                5                0                1                1
10               3                5                1                0                2                0                1
11               3                9                1                9                6                2                2
12               6                7                0                5                5                2                2
13               3                5                1                3                0                0                1
14               6                3                1                1                6                1                1
15               5                3                1                5                5                2                2
16               4                4                2                4                9                1                2
17               0                2                2                8                1                1                2
18               7                7                0                4                4                0                0
19               1                7                1                1                6                1                1
20               5                3                1                8                2                1                1
21               4                0                2                8                3                1                2
22               8                9                0                5                7                2                2
23               9                0                1                4                0                0                1
24               8                3                1                4                1                0                1
25               9                9                0                4                9                1                1
26               3                7                1                9                6                2                2
27               8                4                1                2                0                0                1
28               1                1                2                8                5                2                2
29               9                2                1                5                6                2                2
30               5                2                1                2                9                1                1
> |

```

Gambar 3

```

30                5                1                1                6
>
> print(table(tabel[, "jumlah_yang_dibayar_dollar"]) / length(bilangan_acak_pertama))
      0      1      2
0.2000000 0.4333333 0.3666667

```

Gambar 4

Jika diulang sebanyak 4 kali, maka diperoleh hasil seperti pada Gambar 5.

```

>
> print(table(tabel[, "jumlah_yang_dibayar_dollar"]) / length(bilangan_acak_pertama))
      0      1      2
0.06666667 0.56666667 0.36666667
\

>
> print(table(tabel[, "jumlah_yang_dibayar_dollar"]) / length(bilangan_acak_pertama))
      0      1      2
0.1000000 0.5666667 0.3333333
>

>
> print(table(tabel[, "jumlah_yang_dibayar_dollar"]) / length(bilangan_acak_pertama))
      0      1      2
0.06666667 0.56666667 0.36666667
>

>
> print(table(tabel[, "jumlah_yang_dibayar_dollar"]) / length(bilangan_acak_pertama))
      0      1      2
0.1 0.5 0.4
\

```

Gambar 5

2 Berapa Banyak Simulasi yang Dilakukan?

Jawaban dari pertanyaan ini bergantung pada apa yang sedang kita estimasi dan seberapa akurat kita ingin memperkirakannya (*how accurately we wish to estimate it*). Misalkan kita ingin mengestimasi pembayaran yang diharapkan (*the expected payment*) ke dalam 1% dari nilai sebenarnya (*to within 1% of its true value*) dengan probabilitas 95% (mengestimasi pembayaran yang diharapkan dalam contoh 1). Data dalam Tabel 2 dapat dipandang sebagai hasil dari studi percontohan (*pilot study*). Kita akan menotasikan rata-rata (*mean*) dan varians (*variance*) hasil pembayaran dari satu simulasi (*from a single simulation*) dengan μ dan σ^2 . Berdasarkan *additive rule* untuk rata-rata (*means*) dan varians (*variances*) dari jumlah variabel-variabel acak independen (*according to the additive rules for means and variances of sums of independent random variables*) (telah dijelaskan pada bagian 4.1 dan 4.2), total pembayaran T , terhadap n simulasi-simulasi independen (*in respect of n independent simulations*), akan memiliki rata-rata (*mean*) $n\mu$ dan varians $n\sigma^2$ (*variance*). Selanjutnya oleh teorema limit sentral (*central limit theorem*),

$$\frac{(T - n\mu)}{\sqrt{n\sigma^2}}$$

Akan mendekati distribusi normal (*will be approximately normally distributed*) dengan rata-rata nol (*zero mean*) dan varians satu (*unit variance*), untuk n yang besar (*with zero mean and unit variance for large n*). Simulasi estimasi dari rata-rata μ diberikan oleh $\frac{T}{n}$. Kita membutuhkan (*require*)

$$P\left(0,99\mu < \frac{T}{n} < 1,01\mu\right) = 0,95.$$

Yakni,

$$P(0,99\mu n < T < 1,01\mu n) = 0,95,$$

atau

$$P\left(\frac{-0,01n\mu}{\sqrt{n\sigma^2}} < \frac{T - n\mu}{\sqrt{n\sigma^2}} < \frac{0,01n\mu}{\sqrt{n\sigma^2}}\right) = 0,95.$$

95% titik-titik dari *unit normal distribution* adalah $\pm 1,96$, sehingga

$$\frac{0,01n\mu}{\sqrt{n\sigma^2}} = 1,96$$

di mana

$$n = \left(\frac{1,96\sigma}{0,01\mu}\right)^2.$$

Ekspektasi (*expectation*) μ dan standar deviasi σ tentu saja tidak diketahui (*are of course unknown*). Bagaimanapun, kita mengetahuinya dari studi percontohan (*pilot study*), bahwa rata-rata pembayaran kira-kira/sekitar 1,47 (*the mean payment was about 1,47*); standar deviasi sampel dari pembayaran dapat dihitung secara mudah, yakni 0,56. Perkiraan dari jumlah simulasi yang diperlukan

$$n = \left(\frac{1,96 \times 0,56}{0,01 \times 1,47} \right)^2 = 5500 \text{ (katakanlah).}$$

3 Pembangkit Bilangan Acak (*Random Number Generators*)

Pembangkit bilangan acak adalah metode komputer atau metode fisik (*computerized or physical method*) yang menghasilkan angka-angka/bilangan-bilangan, di mana bilangan-bilangan tersebut tidak memiliki pola (*have no sequential pattern*) dan terbentuk secara acak atau tidak beraturan (terbentuk secara kebetulan / tanpa diatur atur) (*are arranged purely by chance*). Pada awalnya, banyak cara telah diterapkan/digunakan untuk membangkitkan *random variates* (bilangan acak/penyimpangan acak/menghasilkan sesuatu yang acak/secara kebetulan), seperti: melempar dadu (*rolling dice*), melempar koin (*flipping coins*), memutar roda (*roulette wheels*), dan mengocok kartu (*shuffling cards*). Metode fisik ini tidak praktis bila sejumlah besar bilangan acak dibutuhkan dalam praktiknya. Pada tahun 1947, *Rand Corporation* menghasilkan nomor-nomor acak dengan menggunakan perangkat jenis roulette elektronik (*by use of an electronic roulette type device*) yang terhubung ke komputer. Sebuah buku, di mana dalam buku tersebut terdapat angka-angka acak (*A book, with a list of all the numbers*), yang diterbitkan oleh *Rand Corporation*. Angka-angka acak kemudian banyak diterapkan dalam aplikasi statistik (*statistics*), desain eksperimental (*experimental design*), kriptografi (*cryptography*) dan disiplin ilmu lainnya. Bagaimanapun, dengan munculnya komputer berkecepatan tinggi di tahun 1950an (*with the advent of high-speed computers in the 1950s*), algoritma-algoritma matematika menjadi praktis dan perkembangan-perkembangan baru membuat peningkatan cara yang lebih praktis/cepat mengenai bagaimana cara menghasilkan sejumlah besar bilangan acak (Thomopoulos, 2013).

4 *Linear Congruent Generators* (Pembangkit Kongruen Linier)

Pada tahun 1951, Lehmer memperkenalkan suatu cara untuk membangkitkan angka-angka acak, yang mana cara tersebut disebut dengan *linear congruent generator* (LCG). Metode ini mampu ber-adaptasi/menyesuaikan dengan baik terhadap aplikasi-aplikasi komputer dan saat ini paling sering digunakan. Metode ini mengharuskan menggunakan fungsi modulo sebagai mana yang akan diperlihatkan berikut ini. LCG memanggil tiga parameter (a , b , m) dan menggunakan aritmatika modular (*modular arithmetic*). Untuk memperoleh nilai ke- i dari w (*to obtain the i -th value of w*), suatu fungsi menggunakan nilai prior dari w sebagaimana diperlihatkan berikut ini (Thomopoulos, 2013).

$$w_i = (a w_{i-1} + b) \text{ modulo}(m)$$

Pada persamaan di atas, w merupakan urutan/rangkaian dari bilangan bulat (*w is a sequence of integers that are generated from this function*) yang dihasilkan dari fungsi tersebut, dan i merupakan indeks dari urutan/rangkaian. Andaikan $m = 32$, $a = 5$, dan $b = 3$. Juga asumsikan bahwa *seed* (pada $i = 0$) untuk LCG adalah $w_0 = 11$ (Thomopoulos, 2013). Terapkan

$$w_i = (a w_{i-1} + b) \text{ modulo}(m)$$

$$w_i = (5w_{i-1} + 3) \text{ modulo}(32)$$

sehingga menghasilkan barisan bilangan (*sequence*):

26,5,28,15,14,9,16,19,2,13,4,23,22,17,24,27,10,21,12,31,30,25,0,3,18,29,20,7,6,1,8,11

Perhatikan bahwa bilangan-bilangan di atas sebanyak 32 bilangan dari w untuk i dimulai dari 1 sampai 32, dan seluruh bilangan-bilangannya berbeda (*there are 32 values of w for i ranging from 1 to 32, and all are different*). Ini disebut dengan istilah *full cycle* (siklus penuh) dari seluruh nilai-nilai sisa yang mungkin dari 0 sampai 31, ketika $m = 32$. Bilangan terakhir dalam barisan angka tersebut sama dengan *seed* $w_0 = 11$ (*the last number in the sequence is the same as the seed $w_0 = 11$*). Barisan-barisan angka yang kedua yang dihasilkan untuk siklus kedua (*the sequence of numbers that are generated for the second cycle*) akan sama dengan barisan-barisan angka pada siklus pertama karena *the seed has been repeated* (karena *seed* diulang kembali). Pembaca harus lebih berhati-hati bahwa ini tidak mudah untuk menemukan kombinasi dari a dan b yang memberikan *full cycle* (siklus penuh) untuk suatu modulus m . Perhatikan bahwa $m = 32$, $a = 5$ dan $b = 3$ merupakan satu kombinasi yang bekerja (*is one such combination that works*) (Thomopoulos, 2013).

Sekarang perhatikan bahwa parameter-parameter $m = 32$, $a = 7$, dan $b = 13$. Juga asumsikan bahwa *seed* untuk LCG $w_0 = 20$. Dengan menerapkan

$$w_i = (7w_{i-1} + 13) \text{ modulo } (32)$$

menghasilkan barisan-barisan angka sebanyak 32, yakni:

25, 28, 17, 4, 9, 12, 1, 20

25, 28, 17, 4, 9, 12, 1, 20

25, 28, 17, 4, 9, 12, 1, 20

25, 28, 17, 4, 9, 12, 1, 20

Perhatikan bahwa terdapat 8 bilangan yang berulang (terdapat suatu siklus dari 8 bilangan) (*there is a cycle of eight numbers*). Secara umum, ketika satu bilangan dalam barisan-barisan bilangan, nilainya sama dengan *seed*, dalam hal ini $w_0 = 20$ (*when one of the numbers in the sequence is the same as the seed, $w_0 = 20$*), barisan-barisan bilangan akan mengulangi/berulang dengan bilangan-bilangan yang sama, yakni 25-20 (maksudnya 25, 28, 17, 4, 9, 12, 1, 20).

Contoh 4 Membangkitkan Bilangan Acak dari Populasi Berdistribusi Normal dengan R

Fungsi probabilitas dari distribusi normal adalah sebagai berikut (Johnson dan Bhattacharyya, 2011:231, Montgomery dan Runger, 2014:119).

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{untuk } -\infty < x < \infty$$

Perhatikan bahwa:

- ⇒ Pada fungsi probabilitas distribusi normal terdapat 2 parameter, yakni rata-rata μ , di mana $-\infty < \mu < \infty$, dan standar deviasi σ , di mana $\sigma > 0$.
- ⇒ Perhatikan bahwa e merupakan bilangan eksponensial yang didekati dengan bilangan 2,7183, sementara π didekati dengan bilangan 2,7183.

Bentuk dari kurva distribusi normal diperlihatkan pada Gambar 6.

Gambar 6 Kurva Distribusi Normal

Gambar 7 Berbagai Kurva Distribusi Normal

Pada Gambar 7 disajikan 4 kurva distribusi normal, yang masing-masing berwarna kuning, hijau, biru, dan merah.

- ⇒ Pada kurva distribusi normal yang berwarna kuning, nilai rata-rata $\mu = 0$ dan standar deviasi $\sigma = 2,5$.
- ⇒ Pada kurva distribusi normal yang berwarna hijau, nilai rata-rata $\mu = 0$ dan standar deviasi $\sigma = 2$.
- ⇒ Pada kurva distribusi normal yang berwarna hijau, nilai rata-rata $\mu = 0$ dan standar deviasi $\sigma = 1,5$.
- ⇒ Pada kurva distribusi normal yang berwarna merah, nilai rata-rata $\mu = 0$ dan standar deviasi $\sigma = 1$.

Gambar 8 Berbagai Kurva Distribusi Normal

Pada Gambar 8 disajikan 4 kurva distribusi normal, yang masing-masing berwarna kuning, hijau, biru, dan merah.

- ⇒ Pada kurva distribusi normal yang berwarna kuning, nilai rata-rata $\mu = 5$ dan standar deviasi $\sigma = 3$.
- ⇒ Pada kurva distribusi normal yang berwarna hijau, nilai rata-rata $\mu = 1$ dan standar deviasi $\sigma = 3$.
- ⇒ Pada kurva distribusi normal yang berwarna hijau, nilai rata-rata $\mu = 0$ dan standar deviasi $\sigma = 3$.
- ⇒ Pada kurva distribusi normal yang berwarna merah, nilai rata-rata $\mu = -1$ dan standar deviasi $\sigma = 3$.

Misalkan $x = 2$, dengan $\mu = 0$ dan $\sigma = 1$, maka

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \\ f(x=2) &= \frac{1}{\sqrt{2\pi}(1)} e^{-\frac{1}{2}\left(\frac{2-0}{1}\right)^2} \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(2)^2} \\ &= \frac{1}{\sqrt{2\pi}} e^{-2} \\ &= 0,05399097 \end{aligned}$$

Gambar 9.9 merupakan hasil perhitungan berdasarkan R (*R Notebook*).


```

####{r}
x_normal = 2
rata_rata = 0
standar_deviasi = 1
f_x_normal <- (1/sqrt(2*pi*standar_deviasi)) * exp( (-1/2)* ((x_normal-rata_rata)/standar_deviasi)^2 )
f_x_normal

####
[1] 0.05399097

####{r}
dnorm(2,0,1)

####
[1] 0.05399097

```

Gambar 9.9 Hasil Perhitungan dengan R (*R Notebook*)

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 7 dan Gambar 8.

```

library(ggplot2)
xNormal <- seq(-4,4,length=200)
xNormal

standar_deviasi = 1
rata_rata = 0
yNormal <- (1/(standar_deviasi * sqrt(2*pi))) * exp(-((xNormal-
rata_rata)^2)/(2*standar_deviasi^2))
yNormal
zNormal <- dnorm(xNormal, mean=0, sd=1)
zNormal
plot(xNormal,yNormal)
plot(xNormal,yNormal, type="l", lwd=2, col = "blue", xlim = c(-3.5,3.5))
panjang_interval_Normal2 <- (max(xNormal2)-min(xNormal2))/45

distribusi_frekuensi_Normal2 <- cut(xNormal2, breaks =
seq(min(xNormal2),max(xNormal2),by=panjang_interval_Normal2))

jumlah_Normal2 <- table(distribusi_frekuensi_Normal2)

rentang_Normal2 <- names(jumlah_Normal2)

jumlah_Normal2 <- as.vector(jumlah_Normal2)

kelas_interval_Normal2 <- c(1:length(jumlah_Normal2))

```

```
ggplot(data=NULL, aes(x=factor(kelas_interval_Normal2), y=jumlah_Normal2,
fill=rentang_Normal2)) + geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_Normal2/1.3, label=jumlah_Normal2)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_Normal2))))
```

```
curve(dnorm(x, mean=0, sd=1), col="red", xlim = c(-5,5))
curve(dnorm(x, mean=0, sd=1), col="red", xlim = c(-8,8))
curve(dnorm(x, mean=0, sd=1.5), col="blue", xlim = c(-8,8), add = TRUE)
curve(dnorm(x, mean=0, sd=2), col="green", xlim = c(-8,8), add = TRUE)
curve(dnorm(x, mean=0, sd=2.5), col="yellow", xlim = c(-8,8), add = TRUE)
curve(dnorm(x, mean=-1, sd=3), col="red", xlim = c(-15,15))
curve(dnorm(x, mean=0, sd=3), col="blue", xlim = c(-15,15), add = TRUE)
curve(dnorm(x, mean=1, sd=3), col="green", xlim = c(-15,15), add = TRUE)
curve(dnorm(x, mean=5, sd=3), col="yellow", xlim = c(-15,15), add = TRUE)
```

```
5
6
7
8 {r}
9 library(ggplot2)
10 xNormal <- seq(-4,4,length=200)
11 xNormal
12
13 standar_deviasi = 1
14 rata_rata = 0
15 yNormal <- (1/(standar_deviasi * sqrt(2*pi))) * exp(-((xNormal-rata_rata)^2)/(2*standar_deviasi^2))
16 yNormal
17 zNormal <- dnorm(xNormal, mean=0, sd=1)
18 zNormal
19 plot(xNormal,yNormal)
20 plot(xNormal,yNormal, type="l", lwd=2, col = "blue", xlim = c(-3.5,3.5))
21 panjang_interval_Normal2 <- (max(xNormal2)-min(xNormal2))/45
22
23 distribusi_frekuensi_Normal2 <- cut(xNormal2, breaks = seq(min(xNormal2),max(xNormal2),by=panjang_interval_Normal2))
24
25 jumlah_Normal2 <- table(distribusi_frekuensi_Normal2)
26
27 rentang_Normal2 <- names(jumlah_Normal2)
28
29 jumlah_Normal2 <- as.vector(jumlah_Normal2)
30
31 kelas_interval_Normal2 <- c(1:length(jumlah_Normal2))
32
33 ggplot(data=NULL, aes(x=factor(kelas_interval_Normal2), y=jumlah_Normal2, fill=rentang_Normal2)) + geom_bar(stat="identity") +
34 xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") + geom_text(aes(y=jumlah_Normal2/1.3, label=jumlah_Normal2)) +
35 theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3, hjust=0.5)) + labs(fill="Kelas Interval") +
36 scale_fill_manual(values = c(heat.colors(length(jumlah_Normal2))))
37
38 curve(dnorm(x, mean=0, sd=1), col="red", xlim = c(-5,5))
39 curve(dnorm(x, mean=0, sd=1), col="red", xlim = c(-8,8))
40 curve(dnorm(x, mean=0, sd=1.5), col="blue", xlim = c(-8,8), add = TRUE)
```

Gambar 10 Kode R Dibuat dalam R Notebook

Gambar 11 diperlihatkan contoh membangkitkan 10 bilangan acak dari populasi berdistribusi normal dengan rata-rata 0 dan standar deviasi 1.

```
60 >>>{r}
61
62 bilangan_acak_normal_1 = rnorm(n=10, mean = 0, sd = 1)
63 print(bilangan_acak_normal_1)
64
65
66
67 >>>
```

[1] 1.1717533 -0.1410922 1.0244525 -0.4589922 -1.6599425 0.5237002 -0.3427638 1.1004662 -0.1846594 0.5667827

Gambar 11 Membangkitkan 10 Bilangan Acak Berdistribusi Normal

Contoh 5 Membangkitkan Bilangan Acak dari Populasi Berdistribusi t dengan R

Fungsi probabilitas dari distribusi t adalah sebagai berikut (Montgomery dan Runger, 2014:283).

$$f(x) = \frac{\Gamma\left[\frac{(k+1)}{2}\right]}{\sqrt{\pi k} \Gamma\left(\frac{k}{2}\right)} \times \frac{1}{\left[\left(\frac{x^2}{k}\right) + 1\right]^{\frac{(k+1)}{2}}} \text{ untuk } -\infty < x < \infty$$

Perhatikan bahwa:

- ⇒ k merupakan derajat bebas (*degrees of freedom*).
- ⇒ Nilai rata-rata (*mean*) dari distribusi t adalah 0, yakni $\mu = 0$.
- ⇒ Nilai varians (*variance*) dari distribusi t adalah $\frac{k}{k-2}$, untuk $k > 2$, yakni $\sigma^2 = \frac{k}{k-2}$, di mana $k > 2$.

Pada Gambar 12 disajikan 4 kurva, yang masing-masing berwarna hitam, merah, biru, hijau, dan kuning.

Gambar 12 Berbagai Kurva Distribusi t

- ⇒ Kurva yang berwarna hitam merupakan kurva distribusi normal.
- ⇒ Kurva yang berwarna merah, biru, hijau, dan kuning merupakan kurva distribusi t.
- ⇒ Pada kurva distribusi t yang berwarna merah, nilai derajat bebas (*degree of freedom*) adalah 1.
- ⇒ Pada kurva distribusi t yang berwarna biru, nilai derajat bebas (*degree of freedom*) adalah 2.
- ⇒ Pada kurva distribusi t yang berwarna hijau, nilai derajat bebas (*degree of freedom*) adalah 3.
- ⇒ Pada kurva distribusi t yang berwarna kuning, nilai derajat bebas (*degree of freedom*) adalah 25.

Berdasarkan Gambar 12, perhatikan bahwa, semakin besar nilai derajat bebas, maka kurva dari distribusi t akan semakin mendekati kurva dari distribusi normal. Misalkan $x = 2$ dan $k = 100$, maka

$$f(x) = \frac{\Gamma\left[\frac{(k+1)}{2}\right]}{\sqrt{\pi k} \Gamma\left(\frac{k}{2}\right)} \times \frac{1}{\left[\left(\frac{x^2}{k}\right) + 1\right]^{\frac{(k+1)}{2}}}$$

$$f(x=2) = \frac{\Gamma\left[\frac{(100+1)}{2}\right]}{\sqrt{100\pi} \Gamma\left(\frac{100}{2}\right)} \times \frac{1}{\left[\left(\frac{2^2}{100}\right) + 1\right]^{\frac{(100+1)}{2}}}$$

$$= \frac{\Gamma(50,5)}{\sqrt{100\pi} \Gamma(50)} \times \frac{1}{\left[\left(\frac{4}{100}\right) + 1\right]^{50,5}}$$

$$= 0,05490864$$

```

{r}
k=100
x_t = 2
a=gamma((k+1)/2)
b=sqrt(pi*k) * gamma(k/2)
c=((x_t^2)/k)+1)^(k+1)/2)
(a/b)*(1/c)

[1] 0.05490864

{r}
dt(x=2, df=100)

[1] 0.05490864

```

Gambar 13 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 12.

```
xt <- rt(1000, df=3)
xt

panjang_interval_xt <- (max(xt)-min(xt))/45

distribusi_frekuensi_xt<- cut(xt, breaks = seq(min(xt),max(xt),by=panjang_interval_xt))

jumlah_xt <- table(distribusi_frekuensi_xt)
rentang_xt <- names(jumlah_xt)

jumlah_xt <- as.vector(jumlah_xt)

kelas_interval_xt <- c(1:length(jumlah_xt))

ggplot(data=NULL, aes(x=factor(kelas_interval_xt), y=jumlah_xt, fill=rentang_xt)) +
  geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_xt/1.3, label=jumlah_xt)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_xt))))

yt <- dt(xt, df=3)
yt

plot(xt,yt, xlim = c(-7,7))
curve(dt(x, df=3), col="red", xlim = c(-7,7))
curve(dnorm(x), -4, 4, col = "red")
curve(dt(x, df = 1), add = TRUE)
curve(dt(x, df = 2), add = TRUE)
curve(dt(x, df = 3), add = TRUE)

curve(dnorm(x), -3, 3, col = "black")
curve(dt(x, df = 1), col="red", xlim=c(-3,3), ylim=c(0,0.5), add=TRUE)
curve(dt(x, df = 2), add = TRUE, col="blue")
curve(dt(x, df = 3), add = TRUE, col="green")
curve(dt(x, df = 25), add = TRUE, col="yellow")
```

```
59 ~~~{r}
60 kt <- rt(1000, df=3)
61 xt
62
63 panjang_interval_xt <- (max(xt)-min(xt))/45
64
65 distribusi_frekuensi_xt<- cut(xt, breaks = seq(min(xt),max(xt),by=panjang_interval_xt))
66
67 jumlah_xt <- table(distribusi_frekuensi_xt)
68 rentang_xt <- names(jumlah_xt)
69
70 jumlah_xt <- as.vector(jumlah_xt)
71
72 kelas_interval_xt <- c(1:length(jumlah_xt))
73
74
75 ggplot(data=NULL, aes(x=factor(kelas_interval_xt), y=jumlah_xt, fill=rentang_xt)) + geom_bar(stat="identity") +
76   xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") + geom_text(aes(y=jumlah_xt/1.3, label=jumlah_xt)) +
77   theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3, hjust=0.5)) + labs(fill="Kelas Interval") +
78   scale_fill_manual(values = c(heat.colors(length(jumlah_xt))))
79
80 yt <- dt(xt, df=3)
81 yt
82
83 plot(xt,yt, xlim = c(-7,7))
84 curve(dt(x, df=3), col="red", xlim = c(-7,7))
85 curve(dnorm(x), -4, 4, col = "red")
86 curve(dt(x, df = 1), add = TRUE)
87 curve(dt(x, df = 2), add = TRUE)
88 curve(dt(x, df = 3), add = TRUE)
89
90 curve(dnorm(x), -3, 3, col = "black")
91 curve(dt(x, df = 1), col="red", xlim=c(-3,3), ylim=c(0,0.5), add=TRUE)
92 curve(dt(x, df = 2), add = TRUE, col="blue")
93 curve(dt(x, df = 3), add = TRUE, col="green")
94 curve(dt(x, df = 25), add = TRUE, col="yellow")
95
96
97
98
99
100
```

Gambar 14 Kode R Dibuat dalam R Notebook

Gambar 15 diperlihatkan contoh membangkitkan 10 bilangan acak dari populasi berdistribusi t dengan derajat bebas (*degree of freedom*) 9.

```
58
59 ~~~{r}
60
61 bilangan_acak_t = rt(10, df=9)
62 print(bilangan_acak_t)
63
64
65
66
67
```

```
[1] -0.7496665 -0.3003776 -0.1061859  0.2463765 -1.5768103 -1.0484831 -0.6235347  0.8867595 -1.5602300 -0.5352198
```

Gambar 15 Membangkitkan 10 Bilangan Acak Berdistribusi t

Contoh 6 Membangkitkan Bilangan Acak dari Populasi Berdistribusi *Chi-Squared* dengan R

Fungsi probabilitas dari distribusi *chi-squared* adalah sebagai berikut (Walpole dkk., 2007:200).

$$f(x; v) = \begin{cases} \frac{1}{2^{v/2} \Gamma\left(\frac{v}{2}\right)} x^{\frac{v}{2}-1} e^{-\frac{x}{2}} & ; x > 0 \\ 0 & ; x \leq 0 \end{cases}$$

Fungsi probabilitas *chi-squared* memiliki satu parameter tunggal, yakni v , yang mana v merupakan derajat bebas (*degree of freedom*). Derajat bebas v merupakan bilangan bulat positif (*positive integer*). Berbagai bentuk dari kurva distribusi *chi-squared* diperlihatkan pada Gambar 16.

Gambar 16 Berbagai Kurva Distribusi *Chi-Squared*

Misalkan $x = 2$ dan $v = 100$, maka

$$\begin{aligned} f(x) &= \frac{1}{2^{v/2} \Gamma\left(\frac{v}{2}\right)} x^{\frac{v}{2}-1} e^{-\frac{x}{2}} \\ f(x=100) &= \frac{1}{2^{100/2} \Gamma\left(\frac{100}{2}\right)} 2^{\frac{100}{2}-1} e^{-\frac{2}{2}} \\ &= \frac{1}{2^{50} \Gamma(50)} 2^{49} e^{-1} \\ &= 0,007664155 \end{aligned}$$

Gambar 17 diperlihatkan hasil perhitungan dengan R.

```
## {r}
x=2
v=10
a=2^(v/2)
b=gamma(v/2)
c=x^((v/2)-1)
d=exp(-x/2)

(1/(a*b)) * (c*d)

[1] 0.007664155

## {r}
dchisq(x=2, df=10)

[1] 0.007664155
```

Gambar 17 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 16.

```

x_chi_squared <- rchisq(1000, df=5)
x_chi_squared
panjang_interval_chi_squared <- (max(x_chi_squared)-min(x_chi_squared))/45

distribusi_frekuensi_chi_squared<-          cut(x_chi_squared,          breaks          =
seq(min(x_chi_squared),max(x_chi_squared),by=panjang_interval_chi_squared))

jumlah_chi_squared <- table(distribusi_frekuensi_chi_squared)

rentang_chi_squared <- names(jumlah_chi_squared)

jumlah_chi_squared <- as.vector(jumlah_chi_squared)

kelas_interval_chi_squared <- c(1:length(jumlah_chi_squared))

ggplot(data=NULL,      aes(x=factor(kelas_interval_chi_squared),      y=jumlah_chi_squared,
fill=rentang_chi_squared)) + geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_chi_squared/1.3, label=jumlah_chi_squared)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_chi_squared))))
y_chi_squared <- dchisq(x_chi_squared, df=5)
y_chi_squared
plot(x_chi_squared,y_chi_squared, xlim = c(0,20))
curve(dchisq(x, df=5), col="red", xlim = c(0,20))
hist(x_chi_squared, prob=TRUE)
curve( dchisq(x, df=5), col='green', add=TRUE)
curve( dchisq(x, df=10), col='red', add=TRUE )

curve(dchisq(x, df=1), col="red", xlim = c(0,30), ylim=c(0,0.3))
curve(dchisq(x, df=2), col="green", xlim = c(0,30), add=TRUE)
curve(dchisq(x, df=3), col="yellow", xlim = c(0,30), add=TRUE)
curve(dchisq(x, df=4), col="blue", xlim = c(0,30), add=TRUE)
curve(dchisq(x, df=5), col="black", xlim = c(0,30), add=TRUE)
curve(dchisq(x, df=10), col="orange", xlim = c(0,30), add=TRUE)

```

Gambar 18 diperlihatkan contoh membangkitkan 10 bilangan acak dari populasi berdistribusi *chi-squared* dengan derajat bebas (*degree of freedom*) 9.

```

59
60 ~~~{r}
61
62 bilangan_acak_chi_squared = rchisq(10, df=9)
63 print(bilangan_acak_chi_squared)
64
65
66 ~~~
67

```

[1] 5.816821 3.021741 7.676956 10.589832 6.580318 19.329409 8.734815 8.469084 6.577112 2.080141

Gambar 18 Membangkitkan 10 Bilangan Acak Berdistribusi *Chi-Squared*

Contoh 7 Membangkitkan Bilangan Acak dari Populasi Berdistribusi *F* dengan R

Fungsi probabilitas dari distribusi *F* adalah sebagai berikut (Montgomery dan Runger, 2014:408, Walpole dkk., 2007:261).

$$f(x) = \frac{\Gamma\left(\frac{u+v}{2}\right) \left(\frac{u}{v}\right)^{\frac{u}{2}} x^{\left(\frac{u}{2}\right)-1}}{\Gamma\left(\frac{u}{2}\right) \Gamma\left(\frac{v}{2}\right) \left[\left(\frac{u}{v}\right)x + 1\right]^{(u+v)/2}} ; 0 < x < \infty$$

Pada fungsi probabilitas distribusi *F* di atas, *u* merupakan derajat bebas numerator, sementara *v* merupakan derajat bebas denominator. Rata-rata (*mean*) dari distribusi *F* adalah

$$\mu = \frac{v}{v-2} ; v > 2$$

Sementara varians (*variance*) dari distribusi *F* adalah

$$\sigma^2 = \frac{2v^2(u+v-2)}{u(v-2)^2(v-4)} ; v > 4$$

Berbagai bentuk dari kurva distribusi F diperlihatkan pada Gambar 19, Gambar 20 dan Gambar 21.

Gambar 19 Berbagai Kurva Distribusi F

Gambar 20 Berbagai Kurva Distribusi F

Gambar 21 Berbagai Kurva Distribusi F

Misalkan $x = 2$, $u = 50$ dan $v = 50$, maka

$$f(x) = \frac{\Gamma\left(\frac{u+v}{2}\right) \left(\frac{u}{v}\right)^{\frac{u}{2}} x^{\left(\frac{u}{2}\right)-1}}{\Gamma\left(\frac{u}{2}\right) \Gamma\left(\frac{v}{2}\right) \left[\left(\frac{u}{v}\right)x + 1\right]^{(u+v)/2}}$$

$$f(x=2) = \frac{\Gamma\left(\frac{50+50}{2}\right) \left(\frac{50}{50}\right)^{\frac{50}{2}} 2^{\left(\frac{50}{2}\right)-1}}{\Gamma\left(\frac{50}{2}\right) \Gamma\left(\frac{50}{2}\right) \left[\left(\frac{50}{50}\right)2 + 1\right]^{(50+50)/2}}$$

$$f(x=2) = \frac{\Gamma(50)(1)^{25} 2^{(25)-1}}{\Gamma(25)\Gamma(25)[(1)2 + 1]^{(50)/2}}$$

$$f(x=2) = 0,03692757$$

Gambar 22 diperlihatkan hasil perhitungan dengan R.

```

## {r}
u = 50
v = 50
x = 2

a = (gamma((u+v)/2)) * (u/v)^(u/2) * x^((u/2)-1)
b = gamma(u/2) * gamma(v/2) * ((u/v)*x + 1)^((u+v)/2)
a/b
df(x=2, df1=50, df2=50)

```

```

[1] 0.03692757
[1] 0.03692757

```

Gambar 22 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 19 sampai dengan Gambar 21.

```

library(ggplot2)

xF <- rf(1000, df1=5, df2=15)

panjang_intervalF <- (max(xF)-min(xF))/45

distribusi_frekuensiF <- cut(xF, breaks = seq(min(xF),max(xF),by=panjang_intervalF))

jumlahF <- table(distribusi_frekuensiF)

rentangF <- names(jumlahF)

jumlahF <- as.vector(jumlahF)

kelas_intervalF <- c(1:length(jumlahF))

ggplot(data=NULL, aes(x=factor(kelas_intervalF), y=jumlahF, fill=rentangF)) +
  geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlahF/1.3, label=jumlahF)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlahF))))
yF <- df(xF, df1=5, df2=5)

```



```
plot(xF,yF, col="red")
```

```
curve(df(x,df1=5, df2=5), xlim = c(0,6), col="blue")
curve(df(x,df1=5, df2=1), xlim = c(0,4), col="blue", ylim=c(0,1))
curve(df(x,df1=5, df2=3), xlim = c(0,4), col="red", add=TRUE)
curve(df(x,df1=5, df2=15), xlim = c(0,4), col="yellow", add=TRUE)
curve(df(x,df1=5, df2=100), xlim = c(0,4), col="green", add=TRUE)
curve(df(x,df1=3, df2=5), xlim = c(0,4), col="blue", ylim=c(0,1))
curve(df(x,df1=5, df2=5), xlim = c(0,4), col="red", add=TRUE)
curve(df(x,df1=8, df2=5), xlim = c(0,4), col="yellow", add=TRUE)
curve(df(x,df1=30, df2=5), xlim = c(0,4), col="green", add=TRUE)
curve(df(x,df1=5, df2=5), xlim = c(0,3), col="blue", ylim=c(0,2))
curve(df(x,df1=10, df2=10), xlim = c(0,3), col="red", add=TRUE)
curve(df(x,df1=15, df2=15), xlim = c(0,3), col="yellow", add=TRUE)
curve(df(x,df1=30, df2=30), xlim = c(0,3), col="green", add=TRUE)
curve(df(x,df1=50, df2=50), xlim = c(0,3), col="orange", add=TRUE)
```

Gambar 23 diperlihatkan contoh membangkitkan 10 bilangan acak dari populasi berdistribusi F dengan derajat bebas pertama $df1 = 5$, dan derajat bebas kedua $df2 = 6$.

```
59
60 ~~~~{r}
61
62 bilangan_acak_F = rf(n=10, df1=5, df2=6)
63 print(bilangan_acak_F)
64
65
66 ~~~~
67
```

```
[1] 0.1024592 3.6595081 1.8964979 0.2921075 3.5386631 0.9577929 0.1313222 2.4716721 2.4383944
[10] 0.4800122
```

Gambar 23 Membangkitkan 10 Bilangan Acak Berdistribusi F

Contoh 8 Membangkitkan Bilangan Acak dari Populasi Berdistribusi Eksponensial dengan R

Fungsi probabilitas dari distribusi eksponensial adalah sebagai berikut (Montgomery dan Runger, 2014:133).

$$f(x) = \lambda e^{-\lambda x} ; 0 \leq x < \infty$$

Pada fungsi probabilitas distribusi eksponensial, λ merupakan suatu parameter. Perhatikan bahwa e merupakan bilangan eksponensial yang didekati dengan bilangan 2,7183. Bentuk dari kurva distribusi eksponensial diperlihatkan pada Gambar 24 dan Gambar 25.

Gambar 24 Berbagai Kurva Distribusi Eksponensial

Gambar 25 Berbagai Kurva Distribusi Eksponensial

Misalkan $x = 2$ dan $\lambda = 5$, maka

$$\begin{aligned}f(x) &= \lambda e^{-\lambda x} \\f(x = 2) &= (5)e^{-(5)(2)} \\&= 5e^{-10} \\&= 000226996\end{aligned}$$

Gambar 26 diperlihatkan hasil perhitungan dengan R.

```
```{r}
x=2
lambda=5

lambda * exp(-lambda*x)
dexp(x=2, rate=5)
```

[1] 0.0002269996
[1] 0.0002269996
```

Gambar 26 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 24 dan Gambar 25.

```
library(ggplot2)
x_exp <- rexp(1000, rate=3)
x_exp
panjang_interval_exp <- (max(x_exp)-min(x_exp))/45

distribusi_frekuensi_exp <- cut(x_exp, breaks =
seq(min(x_exp),max(x_exp),by=panjang_interval_exp))

jumlah_exp <- table(distribusi_frekuensi_exp)

rentang_exp <- names(jumlah_exp)
```

```

jumlah_exp <- as.vector(jumlah_exp)

kelas_interval_exp <- c(1:length(jumlah_exp))

ggplot(data=NULL, aes(x=factor(kelas_interval_exp), y=jumlah_exp, fill=rentang_exp)) +
  geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_exp/1.3, label=jumlah_exp)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_exp))))
y_exp <- dexp(x_exp, rate=3)
plot(x_exp,y_exp, col="red")
curve(dexp(x,rate=3), xlim = c(0,2.5), col="blue")
curve(dexp(x,rate=3), xlim = c(0,1.5), col="blue")
curve(dexp(x,rate=4), xlim = c(0,1.5), col="green", add=TRUE)
curve(dexp(x,rate=5), xlim = c(0,1.5), col="yellow", add=TRUE)
curve(dexp(x,rate=6), xlim = c(0,1.5), col="orange", add=TRUE)
curve(dexp(x,rate=10), xlim = c(0,1.5), col="red", add=TRUE)
curve(dexp(x,rate=1/2), xlim = c(0,6), col="blue")
curve(dexp(x,rate=1/3), xlim = c(0,6), col="green", add=TRUE)
curve(dexp(x,rate=1/4), xlim = c(0,6), col="yellow", add=TRUE)
curve(dexp(x,rate=1), xlim = c(0,6), col="orange", add=TRUE)
curve(dexp(x,rate=2), xlim = c(0,6), col="red", add=TRUE)

```

Gambar 27 diperlihatkan contoh membangkitkan 10 bilangan acak dari populasi berdistribusi eksponensial dengan $rate = 3$.

```

59
60 ~~~{r}
61
62 bilangan_acak_eksponensial = rexp(10, rate=3)
63 print(bilangan_acak_eksponensial)
64
65
66 ~~~
67

```

```

[1] 0.20887396 0.79498091 1.18404390 0.03235254 0.97749973 0.31949166 0.05346964 0.06182420
[9] 0.27891722 0.84951250

```

Gambar 27 Membangkitkan 10 Bilangan Acak Berdistribusi Eksponensial

Contoh 9 Membangkitkan Bilangan Acak dari Populasi Berdistribusi Gamma dan Erlang dengan R

Fungsi probabilitas dari distribusi Gamma adalah sebagai berikut (Montgomery dan Runger, 2014:140).

$$f(x) = \frac{\lambda^r x^{r-1} e^{-\lambda x}}{\Gamma(r)} ; x > 0$$

Pada fungsi probabilitas distribusi Gamma, λ dan r merupakan parameter dengan $\lambda > 0$ dan $r > 0$. Jika r bilangan bulat, maka fungsi probabilitas tersebut menjadi fungsi probabilitas dari distribusi **Erlang** (Montgomery dan Runger, 2014:140). Bentuk dari kurva distribusi Gamma diperlihatkan pada Gambar 28 dan Gambar 29.

Gambar 28 Berbagai Kurva Distribusi Gamma

Gambar 29 Berbagai Kurva Distribusi Gamma

Misalkan $x = 2$, $\lambda = 4$, dan $r = 3$, maka

$$f(x) = \frac{\lambda^r x^{r-1} e^{-\lambda x}}{\Gamma(r)}$$

$$f(x = 2) = \frac{4^3 2^{3-1} e^{-(4)(3)}}{\Gamma(3)}$$

$$= 0,04293922$$

Gambar 30 diperlihatkan hasil perhitungan dengan R.

```

`{r}
x=2
lambda=4 #scale
r=3 #shape
#Montgomery dan Runger, 2014:140

(lambda^(r) * (x^(r-1)) * exp(-lambda*x)) / gamma(r)

dgamma(x=2, shape=r, scale=1/lambda )
|
[1] 0.04293922
[1] 0.04293922

```

Gambar 30 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 28 dan Gambar 29.

```

x_gamma <- rgamma(n=1000, shape = 3, scale = 0.5)
x_gamma
panjang_interval_gamma <- (max(x_gamma)-min(x_gamma))/45

distribusi_frekuensi_gamma <- cut(x_gamma, breaks =
seq(min(x_gamma),max(x_gamma),by=panjang_interval_gamma))

jumlah_gamma <- table(distribusi_frekuensi_gamma)

rentang_gamma <- names(jumlah_gamma)

jumlah_gamma <- as.vector(jumlah_gamma)

kelas_interval_gamma <- c(1:length(jumlah_gamma))

ggplot(data=NULL, aes(x=factor(kelas_interval_gamma), y=jumlah_gamma,
fill=rentang_gamma)) + geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_gamma/1.3, label=jumlah_gamma)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_gamma))))
y_gamma <- dgamma(x_gamma, shape=3, scale=0.5)
plot(x_gamma,y_gamma, col="red")
curve(dgamma(x,shape=3, scale=0.5), xlim = c(0,6), col="blue")

curve(dgamma(x,shape=3, scale=0.5), xlim = c(0,8), col="blue")

```

```

curve(dgamma(x,shape=4, scale=0.5), xlim = c(0,8), col="red", add=TRUE)
curve(dgamma(x,shape=5, scale=0.5), xlim = c(0,8), col="yellow", add=TRUE)
curve(dgamma(x,shape=6, scale=0.5), xlim = c(0,8), col="orange", add=TRUE)
curve(dgamma(x,shape=12, scale=0.5), xlim = c(0,8), col="green", add=TRUE)
curve(dgamma(x,shape=3, scale=0.5), xlim = c(0,8), col="blue")
curve(dgamma(x,shape=3, scale=0.9), xlim = c(0,8), col="red", add=TRUE)
curve(dgamma(x,shape=3, scale=1), xlim = c(0,8), col="yellow", add=TRUE)
curve(dgamma(x,shape=3, scale=2), xlim = c(0,8), col="orange", add=TRUE)
curve(dgamma(x,shape=3, scale=10), xlim = c(0,8), col="green", add=TRUE)

```

Gambar 31 diperlihatkan contoh membangkitkan 10 bilangan acak dari populasi berdistribusi gamma dengan $shape = 2$ dan $scale = 1/2$.

```

> ```{r}

bilangan_acak_gamma = rgamma(n=10, shape = 2, scale =1/2)
print(bilangan_acak_gamma)

...

[1] 0.8913458 0.1862845 1.1006946 0.6488375 1.3552772 0.7269393 0.1696457 1.9691758 1.1882863
[10] 1.5641499

```

Gambar 31 Membangkitkan 10 Bilangan Acak Berdistribusi Gamma

Contoh 10 Membangkitkan Bilangan Acak dari Populasi Berdistribusi Weibull dengan R

Fungsi probabilitas dari distribusi Weibull adalah sebagai berikut (Montgomery dan Runger, 2014:143).

$$f(x) = \frac{\beta}{\delta} \left(\frac{x}{\delta}\right)^{\beta-1} e^{-\left(\frac{x}{\delta}\right)^{\beta}} ; x > 0$$

Pada fungsi probabilitas distribusi Weibull, δ disebut dengan *scale parameter*, di mana $\delta > 0$, sementara β disebut dengan *shape parameter* dengan $\beta > 0$ (Montgomery dan Runger, 2014:143). Bentuk dari kurva distribusi Weibull diperlihatkan pada Gambar 32 dan Gambar 33.

Gambar 32 Berbagai Kurva Distribusi Weibull

Gambar 33 Berbagai Kurva Distribusi Weibull

Misalkan $x = 2$, $\beta = 3$, dan $\delta = 4$, maka

$$f(x) = \frac{\beta}{\delta} \left(\frac{x}{\delta}\right)^{\beta-1} e^{-\left(\frac{x}{\delta}\right)^{\beta}}$$

$$f(x = 2) = \frac{3}{4} \left(\frac{2}{4}\right)^{3-1} e^{-\left(\frac{2}{4}\right)^3}$$

$$f(x) = 0,1654682$$

Gambar 34 diperlihatkan hasil perhitungan dengan R.

```

> {r}
beta = 3 #shape parameter
delta = 4 #scale parameter
x = 2
)
(beta/delta) * ((x/delta)^(beta-1)) * exp(-(x/delta)^beta)
dweibull(x=2, shape = beta, scale = delta)
>
[1] 0.1654682
[1] 0.1654682

```

Gambar 34 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 32 dan Gambar 33.

Fungsi **rweibull** digunakan untuk membangkitkan bilangan acak dari distribusi Weibull.

```

x_weibull <- rweibull(1000, shape=3, scale = 0.5)
x_weibull
panjang_interval_weibull <- (max(x_weibull)-min(x_weibull))/45

distribusi_frekuensi_weibull <- cut(x_weibull, breaks =
seq(min(x_weibull),max(x_weibull),by=panjang_interval_weibull))

jumlah_weibull <- table(distribusi_frekuensi_weibull)

rentang_weibull <- names(jumlah_weibull)

jumlah_weibull <- as.vector(jumlah_weibull)

```

```

kelas_interval_weibull <- c(1:length(jumlah_weibull))

ggplot(data=NULL, aes(x=factor(kelas_interval_weibull), y=jumlah_weibull,
fill=rentang_weibull)) + geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_weibull/1.3, label=jumlah_weibull)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_weibull))))
y_weibull <- dweibull(x_weibull, shape=3, scale=0.5)
plot(x_weibull,y_weibull, col="red")
curve(dweibull(x,shape=3, scale=0.5), xlim = c(-1,2), col="blue")

curve(dweibull(x,shape=3, scale=0.5), xlim = c(0,1), col="blue", ylim=c(0,10))
curve(dweibull(x,shape=4, scale=0.5), xlim = c(0,1), col="red", add=TRUE, ylim=c(0,10))
curve(dweibull(x,shape=5, scale=0.5), xlim = c(0,1), col="yellow", add=TRUE, ylim=c(0,10))
curve(dweibull(x,shape=6, scale=0.5), xlim = c(0,1), col="orange", add=TRUE, ylim=c(0,10))
curve(dweibull(x,shape=12, scale=0.5), xlim = c(0,1), col="green", add=TRUE, ylim=c(0,10))
curve(dweibull(x,shape=1, scale=0.5), xlim = c(0,1), col="blue", ylim=c(0,3))
curve(dweibull(x,shape=1, scale=1), xlim = c(0,1), col="red", add=TRUE, ylim=c(0,3))
curve(dweibull(x,shape=1, scale=2), xlim = c(0,1), col="yellow", add=TRUE, ylim=c(0,3))
curve(dweibull(x,shape=1, scale=5), xlim = c(0,1), col="orange", add=TRUE, ylim=c(0,3))
curve(dweibull(x,shape=1, scale=10), xlim = c(0,1), col="green", add=TRUE, ylim=c(0,3))

```

Contoh 11 Membangkitkan Bilangan Acak dari Populasi Berdistribusi Lognormal dengan R

Fungsi probabilitas dari distribusi Lognormal adalah sebagai berikut (Montgomery dan Runger, 2014:145-146).

$$f(x) = \frac{1}{x\omega\sqrt{2\pi}} e^{-\left(\frac{(\ln(x)-\theta)^2}{2\omega^2}\right)} ; 0 < x < \infty$$

Parameter dari fungsi distribusi Lognormal adalah θ dan ω^2 (Montgomery dan Runger, 2014:146). Bentuk dari kurva distribusi Lognormal diperlihatkan pada Gambar 35.

```
{r}
curve(dlnorm(x,meanlog=0, sdlog=1), xlim = c(0,1), col="blue")
curve(dlnorm(x,meanlog=1, sdlog=1), xlim = c(0,1), col="red", add=TRUE)
curve(dlnorm(x,meanlog=2, sdlog=1), xlim = c(0,1), col="black", add=TRUE)
curve(dlnorm(x,meanlog=3, sdlog=1), xlim = c(0,1), col="orange", add=TRUE)
```

Gambar 35 Kurva Distribusi Lognormal

Misalkan $x = 2$, $mean \theta = 3$, dan standar deviasi $\omega = 4$, maka

$$f(x) = \frac{1}{x\omega\sqrt{2\pi}} e^{-\left(\frac{(\ln(x)-\theta)^2}{2\omega^2}\right)}$$

$$f(x = 2) = \frac{1}{(2)(4)\sqrt{2\pi}} e^{-\left(\frac{(\ln(2)-3)^2}{2(4)^2}\right)}$$

$$f(x = 2) = 0,04222769$$

Gambar 36 diperlihatkan hasil perhitungan dengan R.

```

{r}
theta = 3
omega = 4
x=2

(1 / (x*omega*sqrt(2*pi))) * exp( -((log(x)-theta)^2) / (2*omega^2)
dlnorm(x=2, meanlog = theta, sdlog=omega)

[1] 0.04222769
[1] 0.04222769

```

Gambar 36 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 35. Fungsi **rlnorm** digunakan untuk membangkitkan bilangan acak dari distribusi lognormal.

```

x_lnorm <- rlnorm(1000, meanlog=0, sdlog=1)
x_lnorm
panjang_interval_lnorm <- (max(x_lnorm)-min(x_lnorm))/45

distribusi_frekuensi_lnorm <- cut(x_lnorm, breaks =
seq(min(x_lnorm),max(x_lnorm),by=panjang_interval_lnorm))

jumlah_lnorm <- table(distribusi_frekuensi_lnorm)

rentang_lnorm <- names(jumlah_lnorm)

jumlah_lnorm <- as.vector(jumlah_lnorm)

kelas_interval_lnorm <- c(1:length(jumlah_lnorm))

ggplot(data=NULL, aes(x=factor(kelas_interval_lnorm), y=jumlah_lnorm, fill=rentang_lnorm)) +
geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
  geom_text(aes(y=jumlah_lnorm/1.3, label=jumlah_lnorm)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_lnorm))))
y_lnorm <- dlnorm(x_lnorm, meanlog=0, sdlog=1)
plot(x_lnorm,y_lnorm, col="red")

```

```
curve(dlnorm(x,meanlog=0, sdlog=1), xlim = c(0,15), col="blue")
```

```
curve(dlnorm(x,meanlog=0, sdlog=1), xlim = c(0,1), col="blue")
```

```
curve(dlnorm(x,meanlog=1, sdlog=1), xlim = c(0,1), col="red", add=TRUE)
```

```
curve(dlnorm(x,meanlog=2, sdlog=1), xlim = c(0,1), col="black", add=TRUE)
```

```
curve(dlnorm(x,meanlog=3, sdlog=1), xlim = c(0,1), col="orange", add=TRUE)
```

Contoh 12 Membangkitkan Bilangan Acak dari Populasi Berdistribusi Beta dengan R

Fungsi probabilitas dari distribusi Beta adalah sebagai berikut (Montgomery dan Runger, 2014:148-149).

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) + \Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} ; \text{ untuk } x \text{ dalam } [0,1]$$

Parameter dari fungsi distribusi Beta di atas adalah α dan β , di mana $\alpha > 0$ dan $\beta > 0$ (Montgomery dan Runger, 2014:148). Bentuk dari kurva distribusi Beta diperlihatkan pada Gambar 37.

Gambar 37 Kurva Distribusi Beta

Misalkan $x = 0,1$, $\alpha = 1,5$, dan $\beta = 0,5$, maka

$$f(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) + \Gamma(\beta)} x^{\alpha-1} (1 - x)^{\beta-1}$$
$$f(x = 0,1) = \frac{\Gamma(1,5 + 0,5)}{\Gamma(1,5) + \Gamma(0,5)} 0,1^{1,5-1} (1 - 0,1)^{0,5-1}$$
$$f(x = 0,1) = 0,2122066$$

Gambar 38 diperlihatkan hasil perhitungan dengan R.

```
##{r}
x=0.1
alpha=1.5
beta=0.5

(gamma(alpha+beta)/(gamma(alpha)*gamma(beta))) * x^(alpha-1) * (1-x)^(beta-1)
dbeta(x=0.1, shape1 = alpha, shape2 = beta)

##

[1] 0.2122066
[1] 0.2122066
```

Gambar 38 Hasil Perhitungan dengan R

Berikut disajikan kode R untuk membuat tampilan seperti pada Gambar 37. Fungsi **rbeta** digunakan untuk membangkitkan bilangan acak dari distribusi beta.

```
x_beta <- rbeta(1000, shape1=5, shape2=1)
x_beta
panjang_interval_beta <- (max(x_beta)-min(x_beta))/45
```

```

distribusi_frekuensi_beta <- cut(x_beta, breaks =
seq(min(x_beta),max(x_beta),by=panjang_interval_beta))

jumlah_beta <- table(distribusi_frekuensi_beta)

rentang_beta <- names(jumlah_beta)

jumlah_beta <- as.vector(jumlah_beta)

kelas_interval_beta <- c(1:length(jumlah_beta))

ggplot(data=NULL, aes(x=factor(kelas_interval_beta), y=jumlah_beta, fill=rentang_beta)) +
geom_bar(stat="identity") +
  xlab("Kelas Interval") + ylab("Jumlah") + ggtitle("Grafik Frekuensi Batang") +
geom_text(aes(y=jumlah_beta/1.3, label=jumlah_beta)) +
  theme(plot.title = element_text(size = 12, face = "bold", colour = "darkblue", vjust=3,
hjust=0.5)) + labs(fill="Kelas Interval") + scale_fill_manual(values =
c(heat.colors(length(jumlah_beta))))
y_beta <- dbeta(x_beta, shape1 = 5, shape2 = 1)
plot(x_beta,y_beta, col="red")
curve(dbeta(x,shape1 = 5, shape2 = 1), xlim = c(0,1), col="blue")
curve(dbeta(x,shape1 = 5, shape2 = 0.1), xlim = c(0,1), col="blue")
curve(dbeta(x,shape1 = 5, shape2 = 0.5), xlim = c(0,1), col="red", add=TRUE)
curve(dbeta(x,shape1 = 5, shape2 = 1), xlim = c(0,1), col="yellow", add=TRUE)
curve(dbeta(x,shape1 = 5, shape2 = 8), xlim = c(0,1), col="black", add=TRUE)
curve(dbeta(x,shape1 = 5, shape2 = 15), xlim = c(0,1), col="orange", add=TRUE)

```

DAFTAR PUSTAKA

1. Agresti, A. dan B. Finlay. 2009. *Statistical Methods for the Social Sciences*, 4th Edition. United States of America: Prentice Hall.
2. Crawley, M. J. 2015. *Statistics An Introduction Using R*, 2nd Edition. John Wiley & Sons.
3. Dalgaard, P. 2008. *Introductory Statistics with R*, 2nd Edition. Springer.
4. Field, A. 2009. *Discovering Statistics Using SPSS*, 3rd Edition. London: Sage.
5. Gardener, M. 2012. *Beginning R, The Statistical Programming Language*. John Wiley & Sons, Inc.
6. James, G., D. Witten, T. Hastie, dan R. Tibshirani. 2014. *An Introduction to Statistical Learning with Applications in R*. Springer.
7. Johnson, R.A. dan G.K. Bhattacharyya. 2011. *Statistics, Principles and Methods*, 6th Edition. John Wiley and Sons, Inc.
8. Kerns, G. J. 2010. *Introduction to Probability and Statistics Using R*.
9. Maindonald, J. dan W. J. Braun. 2010. *Data Analysis and Graphics Using R*, 3rd Edition. Cambridge.
10. Mann, P. S. dan C.J. Lacke. 2011. *Introductory Statistics, International Student Version*, 7th Edition. Asia: John Wiley & Sons, Inc.
11. Montgomery, D. C. dan G. C. Runger. 2011. *Applied Statistics and Probability for Engineers*, 5th Edition. United States of America: John Wiley & Sons, Inc.

12. Montgomery, D. C. dan G. C. Runger. 2014. *Applied Statistics and Probability for Engineers*, 6th Edition. Wiley.
13. Ott, R.L. dan M. Longnecker. 2001. *An Introduction to Statistical Methods and Data Analysis*, 5th Edition. United States of America: Duxbury.
14. Smidth, R. K. dan D. H. Sanders. 2000. *Statistics a First Course*, 6th Edition. United States of America: McGraw-Hill Companies.
15. Spiegel, M. R. dan L. J. Stephens. 2008. *Theory and Problems of Statistics*, 4th Edition. McGRAW-HILL.
16. Thomopoulos, N. T. 2013, *Essentials of Monte Carlo Simulation*, Springer.
17. Walpole, R. E., R. H. Myres, S. L. Myres, dan K. Ye. 2007. *Probability & Statistics for Engineers & Scientists*, 8th Edition. Prentice Hall.
18. I. B. Hossack, J. H. Pollard, B. Zehnwirth, *Introductory Statistics with Applications in General Insurance*,