

Modul Praktikum Perancangan Percobaan

Fardilla Rahmawati, Muhammad Ammar Sahab

24 Januari 2022

Table of contents

Pendahuluan	5
1 Konsep Dasar Perancangan Percobaan	6
1.1 Percobaan	6
1.1.1 Observasi terkadang tidak cukup	6
1.1.2 Prinsip-prinsip percobaan	7
1.1.3 Aplikasikan konsep-konsep tersebut!	8
1.2 Istilah dalam percobaan	8
1.2.1 Satuan percobaan, respons, dan pengamatan	9
1.2.2 Faktor, taraf, perlakuan, dan interaksi	9
1.2.3 Model tetap, acak, dan campuran (aplikasi teori)	10
1.2.4 Rancangan perlakuan, pengendalian lingkungan, dan respon	10
1.2.5 Aplikasi: studi-studi kasus	11
1.3 Percobaan ke uji: perancangan percobaan	12
1.3.1 Penentuan ulangan	12
1.4 Pengayaan	14
2 Percobaan Faktor Tunggal	16
2.1 Pengendalian lingkungan secara garis besar	16
2.1.1 RAL: unit percobaan homogen	16
2.1.2 Rancangan yang melakukan pengendalian lingkungan	16
2.2 RAL	17
2.2.1 Pengacakan	18
2.2.2 Model	24
2.2.3 Statistik	25
2.2.4 Dekomposisi Jumlah Kuadrat	26
2.2.5 Asumsi Galat	26
2.2.6 Hipotesis dan ujinya	27
2.2.7 Review MKT	27
2.2.8 Review Asumsi-Asumsi	28
2.2.9 Tabel ANOVA	29
2.2.10 Praktek menggunakan R	29
2.2.11 Analisis Percobaan di Python	32
2.3 RAKL	35
2.3.1 Pengacakan	35

2.3.2	Model Linear Aditif	39
2.3.3	ANOVA	40
2.3.4	Analisis dengan R dan Python	41
2.3.5	Topik-topik tambahan	44
2.4	RBSL	56
2.4.1	Pengacakan	56
2.4.2	Model	60
2.4.3	ANOVA	60
2.4.4	Implementasi R dan Python	62
2.4.5	Topik Tambahan	64
3	Perbandingan Nilai Tengah Perlakuan	65
3.1	LSD/BNT	65
3.2	BNJ	68
3.3	Uji Duncan	73
3.4	Kontras	75
3.4.1	Menyusun kontras ortogonal	78
3.4.2	Menghitung Kontras	81
3.5	Polinomial Orthogonal	83
4	Uji Asumsi	88
4.1	Galat percobaan menyebar normal	88
4.1.1	Prosedur Eksploratif	88
4.1.2	Uji formal untuk galat percobaan menyebar normal	94
4.1.3	Konsekuensi dari ketidaknormalan pada galat.	99
4.2	Kehomogenan ragam	100
4.2.1	Prosedur Eksploratif	101
4.2.2	Uji formal untuk kehomogenan ragam	108
4.2.3	Uji Levene dan Brown-Forsythe	111
4.2.4	Plot vs. uji	114
4.3	Galat percobaan saling bebas	116
4.3.1	Prosedur Eksploratif	117
4.3.2	Uji formal untuk galat percobaan saling bebas	119
4.3.3	Konsekuensi dari ketidakbebasan pada galat	125
4.4	Aditivitas	126
4.4.1	Plot Interaksi	127
4.4.2	Uji formal	127
5	Rancangan Faktorial	132
5.1	Faktorial RAL	132
5.1.1	Pembuatan bagan percobaan	132
5.1.2	Model Linear Aditif	136
5.1.3	ANOVA dengan R	138

5.1.4	Tabel dan Plot Interaksi	141
5.1.5	Uji lanjut	145
5.2	Faktorial RAKL	149
5.2.1	Pengacakan	149
5.2.2	Model Linear Aditif	152
5.2.3	ANOVA dengan R	154
6	Rancangan Petak Terpisah (Split Plot Design)	159
6.1	Split Plot RAL	160
6.1.1	Pengacakan	160
6.1.2	ANOVA	166
6.1.3	Analisis menggunakan R	167
6.2	Split Plot RAK	172
6.2.1	Pengacakan	173
6.2.2	ANOVA	177
6.2.3	Analisis menggunakan R	179
6.3	Latihan Soal	182
7	Rancangan Split Blok	184
7.1	Pengacakan	184
7.1.1	Pengacakan di agricolae	184
7.1.2	Pengacakan di edibble	185
7.2	ANOVA	188
7.2.1	Teori	188
7.2.2	Analisis dengan R	190
7.2.3	Latihan: Huasahuasi	196
8	ANCOVA	198
8.1	Data	198
8.2	Visualisasi ANCOVA	199

Pendahuluan

Buku ini akan mengandung materi-materi praktikum Metode Perancangan Percobaan. Materi satu pertemuan akan dimasukkan di satu bab¹. Tiap bab akan mengandung bagian teori dan kode di R dan LaTeX. Slide dan recording praktikum disimpan di [Google Drive](#) dan materi-materi kuliah disimpan di [LMS IPB](#).

¹misal, Bab 5 dari buku ini akan membahas pertemuan 5.

1 Konsep Dasar Perancangan Percobaan

Di bab ini, kita akan membedakan suatu percobaan dengan penelitian observasional dan prinsip dari percobaan tersebut, mengilustrasikan istilah-istilah dalam percobaan, menjelaskan konsep perancangan percobaan, dan mencoba beberapa metode penghitungan ukuran sampel.

1.1 Percobaan

1.1.1 Observasi terkadang tidak cukup

Sebuah penelitian mungkin dimulai dengan **mengamati** dunia sekitar, atau observasi. Mungkin kalian mengamati bahwa pedagang dengan harga murah memiliki lebih banyak pelanggan, atau kalian mengamati bahwa mahasiswa yang lebih banyak belajar memiliki nilai yang lebih tinggi. Namun, kadang hubungan yang ditemukan dalam suatu observasi lebih kompleks dari yang kita amati.



Figure 1.1: Kapal layar dan nahkodanya. Foto dari Ludomil Sawicki (Unsplash).

Bayangkan ada seorang nahkoda¹. Kapal yang dikendalikan oleh nahkoda tersebut berada di tengah angin yang membelokkan kapal tersebut ke kanan dan kiri. Anggap bahwa nahkoda tersebut ingin kapal berlayar lurus, dan nahkoda ini sudah sangat berpengalaman sehingga dia dapat membelokkan kapalnya dengan tepat. Seorang pengamat akan melihat nahkoda tersebut membelokkan kemudi kapalnya ke kanan dan ke kiri, tetapi kapal tetap lurus. Pengamat yang tidak memiliki pengetahuan tentang kapal mungkin menyimpulkan bahwa kemudi tidak berpengaruh pada laju kapal!

Kasus ini merupakan contoh **galat sistematis**. Kita ingin mengetahui pengaruh arah belok kapal ke arah laju kapal, tetapi Nahkoda tidak membelokkan kapal secara acak, melainkan sesuai dengan kondisi angin agar kapal tetap melaju lurus.

1.1.2 Prinsip-prinsip percobaan

Lalu, bagaimana kita dapat mengetahui hubungan antara kemudi kapal dan laju kapal?

Pertama, mungkin kita dapat menyuruh nahkoda untuk membelokkan kemudinya secara acak. Sebuah koin dapat dilempar, dan sisi yang muncul² menentukan pembelokan kapal. Ini adalah **pengacakan** - sebelumnya, jika terjadi angin tertentu, nahkoda pasti membelokkan kapal berkebalikan dengan pengaruh angin tersebut. Peluang kemudi diputar ke kanan dan kiri tidak seimbang. Sekarang, peluang tersebut sama, apapun kondisinya.

Kedua, mungkin ada hal-hal lain yang menyebabkan berubahnya hubungan kemudi dengan laju kapal. Mungkin kapal yang berbeda memiliki sistem kemudi yang berbeda, atau nahkoda berbeda memiliki kemampuan mengemudi berbeda. Oleh karena itu, kita ingin **mengendalikan lingkungan**: kapal homogen, nahkoda homogen. Atau, kita dapat mengulangi percobaan tersebut di kombinasi kapal dan nahkoda berbeda. Rancangan ini dapat disebut Rancangan Bujur Sangkar Latin, yang mengendalikan keragaman dari dua arah. Rancangan Acak Kelompok Lengkap mengendalikan keragaman dari satu arah, dan rancangan pengendalian lingkungan lainnya yang lebih kompleks dapat mengendalikan keragaman dari berbagai arah³. Namun, rancangan **pengendalian lingkungan** tersebut tidak perlu dipakai jika kondisi unit percobaan sudah **homogen** - tidak ada sumber keragaman dari arah manapun.

Ketiga, ada **variabilitas** di percobaan ini. Kita tidak dapat mengobservasi semua sumber keragaman. Teori-teori yang dimiliki tentang cara dunia bekerja selalu berubah - beberapa waktu lalu, orang menganggap penyakit disebabkan oleh cairan tubuh (misal, darah) yang tidak seimbang, bukan jasad-jasad seperti bakteri dan virus. Kemungkinan bahwa teori yang paling mutakhir sekalipun belum dapat mengidentifikasi semua sumber keragaman sangat mungkin benar. Bahkan jika ada faktor yang dapat kita observasi, belum tentu semua faktor tersebut

¹Contoh ini dijelaskan di buku [Causal Inference: The Mixtape](#) buatan Scott Cunningham

²Misal jika muncul uang belok kanan, dan sebaliknya.

³Rancangan Greko-Latin, misal, dapat mengendalikan tiga sumber keragaman

dapat dikendalikan. Mungkin, misal, tiba-tiba muncul ikan besar yang menabrak kapal saat belok, sehingga lajunya berubah⁴. Oleh karena itu, laju kapal pasti tetap beragam.

Agar keragaman tersebut dapat diukur, perlu lebih dari satu pengamatan. Kalau hanya ada satu pengamatan, simpangan baku akan menjadi nol. Selain itu, makin banyak ulangan, makin kecil ragam sebaran contoh rata-rata⁵.

Ini adalah inti percobaan: kita mengumpulkan respon (laju kapal) dari individu yang kondisikan tertentu (nahkoda yang disuruh belok kanan atau kiri). Kita melakukan pengacakan, kontrol lingkungan, dan ulangan.

1.1.3 Aplikasikan konsep-konsep tersebut!

Setelah mengetahui perbedaan observasi dan prinsip dari percobaan, coba pikirkan:

1. Masalah apa yang mungkin muncul jika kita lakukan observasi dalam kasus harga dagangan dan waktu belajar mahasiswa? Mengapa masalah itu terjadi? Bagaimana Anda akan membuat percobaan untuk mengatasi masalah tersebut?
2. Ambil satu hal yang dapat diteliti di sekitar Anda. Apa yang Anda dapat simpulkan dari observasi? Apa yang Anda **tidak** dapat simpulkan dari observasi? Mengapa? Bagaimana Anda akan membuat percobaan untuk meneliti hal tersebut.
3. Buat *cause and effects diagram* dari kasus yang Anda ambil.

1.2 Istilah dalam percobaan

Untuk memperkuat pengertian mengenai istilah percobaan yang Anda miliki, perhatikan contoh tersebut:

Seorang peneliti agronomi melakukan percobaan pada tanaman jagung varietas Arjuna. Jarak tanam di tiap petak diatur berbeda-beda, yaitu 20 x 30 cm², 30 x 30 cm², dan 30 x 40 cm². Jenis pupuk yang diberi selama penelitian adalah pupuk campuran NPK dengan dosis 100, 200, 300, dan 400 kg/ha. Untuk semua unit percobaan dilakukan penyiangan sebanyak 2 kali yaitu pada umur 3 dan 5 minggu setelah tanam.

— Mattjik dan Sumertajaya (2002)

⁴Kecuali Anda bisa berbicara dengan ikan, hal ini susah dikontrol

⁵Ragam populasi adalah σ^2 , sedangkan ragam sebaran contoh rata-rata $\frac{\sigma^2}{n}$.

1.2.1 Satuan percobaan, respons, dan pengamatan

Satuan percobaan dalam kasus ini adalah **petak**. Ingat bahwa satuan percobaan adalah unit terkecil yang diberi perlakuan. Tentu, jarak tanam hanya dapat ditentukan jika ada beberapa tanaman yang disusun di suatu petak; jika hanya ada satu tanaman, tidak mungkin dihitung jarak tanam. Dengan logika yang sama, sangat susah untuk melakukan pemupukan pada satu tanaman tertentu. Pupuk mungkin terbawa ke tanaman lain di petak yang sama oleh angin, air, atau faktor alam lain. Oleh karena itu, unit terkecil yang diberi perlakuan adalah petak.

Satuan pengamatan tergantung tujuan dari apa yang ingin **diamati**. Perubahan apa yang kita ingin amati? Misal **respons** kita adalah produktivitas. Produktivitas mungkin diukur dengan membandingkan berat jagung yang dihasilkan oleh petak tertentu - misal dengan satuan kg/ha. Maka, **satuan pengamatan** adalah petak, karena **respons** diukur di petak. Namun, jika kita ingin mengamati tinggi tanaman, maka yang diukur adalah tanaman, sehingga **satuan pengamatan** adalah tanaman.

1.2.2 Faktor, taraf, perlakuan, dan interaksi

Faktor dalam kasus ini adalah jarak tanam dan pupuk. Varietas dan penyiangan bukan faktor. Mengapa? Jarak tanam dan pupuk **berbeda** di tiap petak (satuan percobaan), tetapi semua petak ditanam varietas sama dan diberi penyiangan sama. Dalam menentukan faktor, pertanyaan yang penting adalah apa yang akan berbeda di tiap satuan percobaan.

Taraf adalah nilai dari faktor yang dicobakan. Dalam kasus ini, ada tiga jarak tanam berbeda ($20 \times 30 \text{ cm}^2$, $30 \times 30 \text{ cm}^2$, $30 \times 40 \text{ cm}^2$) yang merupakan **taraf** dari **faktor** jarak tanam. Ada empat dosis pupuk berbeda (100 kg/ha, 200 kg/ha, 300 kg/ha, 400 kg/ha) yang menjadi taraf faktor dosis pupuk.

Perlakuan adalah suatu prosedur atau metode yang diterapkan pada satuan percobaan. Asumsikan bahwa peneliti ingin menerapkan semua kombinasi taraf faktor pada satuan percobaan. Anggap d_i adalah jarak tanam ke- i dan p_j adalah dosis pupuk ke- j (misal $p_{j=1}$ adalah 100 kg/ha, $p_{j=2}$ adalah 200 kg/ha, dan seterusnya). Oleh karena itu, kita dapat menyusun (di LaTeX):

```
$$
\begin{bmatrix}
d_{\{1\}p_{\{1\}}} & d_{\{2\}p_{\{1\}}} & d_{\{3\}p_{\{1\}}}\backslash\backslash \\
d_{\{1\}p_{\{2\}}} & d_{\{2\}p_{\{2\}}} & d_{\{3\}p_{\{2\}}}\backslash\backslash \\
d_{\{1\}p_{\{3\}}} & d_{\{2\}p_{\{3\}}} & d_{\{3\}p_{\{3\}}}\backslash\backslash \\
d_{\{1\}p_{\{4\}}} & d_{\{2\}p_{\{4\}}} & d_{\{3\}p_{\{4\}}}\backslash\backslash \\
\end{bmatrix}
$$
```

$$\begin{bmatrix} d_1p_1 & d_2p_1 & d_3p_1 \\ d_1p_2 & d_2p_2 & d_3p_2 \\ d_1p_3 & d_2p_3 & d_3p_3 \\ d_1p_4 & d_2p_4 & d_3p_4 \end{bmatrix}$$

Kita temukan bahwa jumlah perlakuan adalah $3 \cdot 4 = 12$, dan tiap perlakuan diberikan dalam matriks tersebut. Jika kita mengetahui **ulangan**, kita dapat menghitung **jumlah unit percobaan** dengan mengalikan perlakuan dan ulangan.

Anggap kita temukan bahwa pengaruh dosis pupuk selalu sama di tiap jarak tanam yang berbeda. Situasi ini berarti tidak ada **interaksi**. Namun, mungkin pengaruh dosis pupuk akan berbeda di jarak tanam yang berbeda - mungkin pengaruh pupuk lebih kecil di jarak tanam yang kecil karena tanaman berkompetisi mencari nutrisi. Kasus ini disebut **interaksi**, di mana pengaruh suatu level dari faktor berbeda jika level faktor lain berbeda.

1.2.3 Model tetap, acak, dan campuran (aplikasi teori)

Anggap Anda bertanya ke peneliti mengenai cara dia memilih taraf dari dosis pupuk dan jarak tanam. Model apa yang tepat berdasarkan jawaban peneliti?

1. Dosis pupuk dan jarak tanam dipilih oleh peneliti secara langsung⁶.
2. Peneliti melakukan proses pengacakan dari semua nilai dosis pupuk dan jarak tanam yang ada (misal dengan mengocok dan mengambil kertas berisi nilai dosis dan jarak tertentu)⁷.
3. Peneliti mengacak nilai jarak tanam, tapi dosis pupuk dipilih sendiri⁸.

1.2.4 Rancangan perlakuan, pengendalian lingkungan, dan respon

Dalam penjelasan sebelumnya, kita telah menyentuh rancangan perlakuan dengan mengasumsikan bahwa peneliti ingin tahu pengaruh kombinasi tiap taraf dari kedua faktor. Ini disebut **rancangan bersilang** karena taraf dari tiap faktor saling bebas. Namun, bisa jadi beberapa dosis pupuk tertentu tidak bisa dipakai di jarak tanam kecil. Misal, untuk melakukan pemupukan 400 kg/ha perlu mesin tertentu yang terlalu besar di petak dengan jarak tanam $20 \times 30 \text{ cm}^2$. Taraf dari faktor tidak saling bebas, sehingga dia **bersarang**. Taraf-taraf tertentu

⁶Model tetap.

⁷Model acak.

⁸Model campuran

bersarang pada taraf lain, misal pupuk 400 kg/ha hanya bersarang di jarak tanam besar, tidak bisa pergi ke jarak tanam kecil.

Lalu, bagaimana jika petak-petak tersebut memiliki kesuburan yang berbeda? Misal sebagian dari petak berada di hutan, sebagian lagi berada di dekat sungai, sebagian berada di gunung. Tentu, kondisi petak sebagai unit percobaan harus dikendalikan agar homogen - ini adalah **rancangan pengendalian lingkungan**.

Terakhir, misal Anda ingin mengukur tinggi tanaman. Apakah Anda ingin mengukur menggunakan penggaris? Meteran? Software? Jawaban pertanyaan ini adalah **rancangan respon**.

1.2.5 Aplikasi: studi-studi kasus

Temukan faktor, level, perlakuan, pengendalian lingkungan, satuan percobaan dan satuan pengamatan, jumlah ulangan, serta respon dari percobaan. Soal ini diambil dari Mattjik dan Sumertajaya (2002):

1. Suatu Lembaga Penelitian Padi melakukan percobaan untuk mengetahui pengaruh penggunaan berbagai pestisida cair terhadap produksi padi. Percobaan dilakukan pada satu hamparan sawah yang mempunyai tingkat kesuburan atau kandungan bahan organik yang relatif sama. Masing-masing perlakuan diberikan pada lahan seluas 4 x 5 m². Perlakuan yang dicobakan ada 7, yaitu Dol-Mix (1 kg), Dol-Mix (2 kg), DDT + γ -BHC, Azodrin, Dimecron-Boom, Dimecron-Knap, dan tanpa pestisida (sebagai kontrol), masing-masing diulang 4 kali. Respon yang diukur adalah produksi gabah per kg/ha. Hasil dari setiap petakan dikonversi ke dalam kg/ha.
2. Percobaan lain pada Balai Penelitian Padi tersebut adalah ingin mengetahui pengaruh kerapatan benih terhadap produksi padi (kg/ha) pada varietas IR8. Kerapatan benih yang dicobakan adalah 25 kg benih/ha, 50 kg benih/ha, 75 kg benih/ha, 100 kg benih/ha, 125 kg benih/ha, 150 kg benih/ha. Lahan yang digunakan di perbukitan (lereng bukit), sehingga tingkat kesuburan tidak sama. Untuk itu petak dibagi jadi 4 kelompok. Masing-masing benih ditanam pada lahan seluas 5 x 5 m² serta hasilnya dikonversi ke satuan kg/ha.
3. Sebuah balai penelitian ingin mengetahui pengaruh varietas dan dosis pupuk terhadap tinggi tanaman tomat (cm). Dosis pupuk yang digunakan adalah 15, 30, dan 45. Jenis varietas yang digunakan adalah varietas 1, 2, dan 3. Tiap perlakuan diulang sebanyak 4 kali. Selain kombinasi perlakuan yang diberikan, semua kondisi relatif homogen. Setiap petak terdiri dari 10 tanaman tomat.
4. Terdapat 3 jenis material untuk pembuatan baterai (A, B, C) yang dicobakan pada 3 temperatur (15 oF, 70 oF, 125 oF). Masing-masing perlakuan diulang sebanyak 3 kali. Dari percobaan tersebut ingin diketahui apakah jenis material dan suhu mempengaruhi daya tahan baterai? Apakah jenis material tertentu cocok untuk suhu tertentu?

1.3 Percobaan ke uji: perancangan percobaan

Kita telah melihat dua contoh percobaan (nahkoda dan varietas jagung). Bagaimana kita menarik kesimpulan dari percobaan tersebut? Suatu **hipotesis** mengenai keadaan dunia (misal, bahwa pengaruh faktor ke respon sebesar β) harus diuji. Fungsi **perancangan percobaan** adalah pembuatan dan pengujian hipotesis tersebut.

1.3.1 Penentuan ulangan

Ada tiga metode penentuan ulangan:

1. Minimal 3
2. Minimal derajat bebas galat 15
3. Menggunakan formula di bawah:

$$r \geq 2 \left(Z_{\alpha/2} + Z_{\beta} \right)^2 \left(\frac{\sigma}{\delta} \right)^2$$

\$\$

$$r \geq 2 \left(Z_{\alpha/2} + Z_{\beta} \right)^2 \left(\frac{\sigma}{\delta} \right)^2$$

\$\$

Metode 2 dan 3 berkaitan erat dengan pengujian suatu hipotesis. Perhatikan bahwa:

1. Uji-F yang digunakan di dalam ANOVA berupa rasio kuadrat tengah: KT perlakuan/KT galat. Sedangkan, kuadrat tengah galat adalah jumlah kuadrat galat dibagi derajat bebas galat. Pada saat kita memberi jumlah minimal pada derajat bebas galat, secara tidak langsung kita memengaruhi besar kuadrat tengah galat, yang memengaruhi statistik uji-F.
2. Formula (3) sangat berkaitan dengan uji. α adalah peluang menolak H_0 jika H_0 benar dalam suatu uji. Sedangkan β adalah peluang menerima H_0 jika H_1 benar⁹. Selain itu, δ adalah perbedaan rata-rata yang diinginkan. Benang merah yang dapat diambil dari formula ini adalah menentukan jumlah ulangan, jika dalam menggunakan uji dapat mendeteksi perbedaan rata-rata sebesar δ dengan peluang kesalahan sebesar α dan β , jika diketahui ragam populasi σ^2 .

⁹Sehingga H_0 salah.

Bagaimana kita menghitung nilai tersebut di R? Untuk formula 2, anggap kita menghitung RAL, sehingga rumus db galat adalah $t(r - 1)$. Oleh karena itu:

$$\begin{aligned} t(r - 1) &\geq 15 \\ r - 1 &\geq \frac{15}{t} \\ r &\geq \frac{15}{t} + 1 \end{aligned}$$

```
$$
\begin{aligned}
&t(r-1)\geq 15\\
&r-1\geq \frac{15}{t}\\
&r\geq \frac{15}{t}+1
\end{aligned}
$$
```

Anggap ada empat perlakuan:

```
t=4
#ceiling dipakai untuk membulatkan ke atas
ceiling(15/t+1)
```

[1] 5

Bagaimana cara kita menghitung formula ketiga? Selain secara manual, dapat dibuat fungsi di R:

```
jmlUlangan<-function(alpha,beta,popvar,delta)
{
  #Temukan nilai Z. Note bahwa alpha dibagi 2
  #lower.tail=T akan membuat nilai Z negatif, karena akan menghitung kuantil dari kiri.
  Za=qnorm(alpha/2,lower.tail=F)
  Zb=qnorm(beta,lower.tail=F)
  r=2*((Za+Zb)^2)*(popvar/delta)^2
  return(ceiling(r))
}
jmlUlangan(0.05,0.10,3,5)
```

[1] 8

```
jmlUlangan(0.05,0.10,0.3,3)
```

[1] 1

Kalian dapat mengubah argumen dari fungsi tersebut untuk mencoba kombinasi nilai ukuran sampel yang berbeda. Note, hasil-hasil perhitungan dibulatkan ke atas karena:

1. Jumlah ulangan pasti bilangan bulat. Perlakuan tidak mungkin diulang 3.5 kali, misal.
2. Semua metode tersebut mensyaratkan **minimal**. Dalam kata lain, jumlah ulangan lebih dari sama dengan hasil rumus tersebut. Jika kita membulatkan saja, ada risiko angka tersebut dibulatkan menjadi angka yang lebih kecil dari hasil rumus. Oleh karena itu, pembulatan ke atas memastikan angka yang dibulatkan lebih besar atau sama dengan hasil rumus yang digunakan.

1.4 Pengayaan

Ukuran sampel juga dapat dihitung dengan fungsi `power.anova.test` di R. Power adalah kuasa uji, yang didefinisikan sebagai probabilitas suatu hipotesis nol ditolak jika hipotesis nol tersebut memang salah. Masukkan jumlah kelompok, ragam antara kelompok, ragam dalam kelompok, kuasa uji, dan taraf nyata yang diinginkan. Jumlah sampel diasumsikan NULL.

```
power.anova.test(groups = 4,  
                  between.var = 3, within.var = 5,  
                  power=0.90, sig.level=0.05,  
                  n=NULL)
```

Balanced one-way analysis of variance power calculation

```
groups = 4  
n = 8.91936  
between.var = 3  
within.var = 5  
sig.level = 0.05  
power = 0.9
```

NOTE: n is number in each group

Jika kita memiliki rata-rata per kelompok, rata-rata tersebut dapat dimasukkan:

```
ratak1 <- c(550, 598, 598, 646)
p <- power.anova.test(groups = length(ratak1),
                      between.var = var(ratak1), within.var = 6400,
                      power=0.9,sig.level=0.05,
                      n=NULL)

p
```

Balanced one-way analysis of variance power calculation

```
groups = 4
n = 20.68861
between.var = 1536
within.var = 6400
sig.level = 0.05
power = 0.9
```

NOTE: n is number in each group

2 Percobaan Faktor Tunggal

Bab ini akan membahas percobaan faktor tunggal dalam RAL, RAK, dan RBSL. Oleh karena itu, **keragaman yang disebabkan perlakuan merupakan level-level dari satu faktor** tertentu saja. Kita akan memulai dengan menjelaskan rancangan-rancangan pengendalian lingkungan secara garis besar. Inti dari bab ini adalah penjelasan dan implementasi tiap rancangan. Di akhir bab, akan ada pengayaan berupa rancangan Greko-Latin dan *Balanced Incomplete Block Designs*.

2.1 Pengendalian lingkungan secara garis besar

Dalam meneliti efek suatu peubah pada peubah lainnya, sering muncul sumber keragaman lain - kita dapat sebut hal ini sebagai faktor **pengganggu**. Ada beberapa faktor pengganggu yang kita tidak dapat observasi dan tidak dapat kendalikan. Dampak faktor-faktor tersebut diatasi oleh pengacakan. Ada beberapa faktor pengganggu yang dapat diobservasi, tapi tidak dapat dikendalikan. ANCOVA, yang akan dipelajari di sesi UAS, mengatasi masalah tersebut. Terakhir, pengganggu yang dapat dikendalikan diatasi dengan rancangan pengendalian lingkungan (*local control*).

2.1.1 RAL: unit percobaan homogen

Rancangan Acak Lengkap digunakan jika semua satuan percobaan homogen. Karena tidak ada sumber keragaman, perlakuan diacak langsung ke unit percobaan.

2.1.2 Rancangan yang melakukan pengendalian lingkungan

Jika satuan percobaan heterogen, ada beberapa rancangan yang dapat mengendalikan faktor pengganggu tersebut. Rancangan yang dipakai tergantung **jumlah arah keragaman**, yang sama saja dengan **jumlah faktor pengganggu yang dapat dikontrol**. Misal, dua arah keragaman berarti ada dua faktor pengganggu yang dapat dikendalikan. Rancangan-rancangan tersebut adalah:

- Rancangan Acak Kelompok Lengkap (RAKL) - **keragaman satu arah**
 - Perlakuan diacak di tiap kelompok

- Syarat: setiap perlakuan muncul sekali dalam setiap kelompok
- Rancangan Bujur Sangkar Latin (RBSL) - **keragaman dua arah**
 - Pengelompokan per baris dan kolom
 - Syarat: setiap perlakuan hanya muncul sekali pada arah baris dan hanya muncul sekali pada arah kolom
- Rancangan Greko Latin - **keragaman tiga arah**
 - Dua Bujur Sangkar Latin digabung: bujur sangkar pertama memiliki perlakuan berhuruf Latin, bujur sangkar kedua memiliki huruf Yunani
 - Syarat: tiap huruf Yunani hanya muncul sekali bersama dengan tiap huruf Latin.
- Rancangan lain yang lebih kompleks, tetapi rancangan tersebut di luar cakupan mata kuliah ini.

2.2 RAL

Setelah melihat penurunan ANOVA untuk Rancangan Acak Lengkap, atau dapat disebut Rancangan Teracak Lengkap (RTL), atau *Completely Randomized Design* (CRD). Karakteristik RAL (selain karakteristik umum percobaan faktor tunggal di awal bab) adalah:

1. Kondisi unit percobaan yang digunakan relatif homogen
2. Karena unit percobaan harus homogen, umumnya percobaan dilakukan di laboratorium atau lingkungan yang dapat dikendalikan
3. Karena lingkungan percobaan terbatas, unit percobaan tidak cukup besar dan jumlah perlakuan terbatas

Dari karakteristik tersebut, kelebihan dalam menggunakan RAL adalah sebagai berikut:

1. Analisis yang digunakan sederhana karena hanya dihitung pengaruh satu faktor
2. Bagan rancangan percobaan lebih mudah karena pengacakan langsung dilakukan ke unit percobaan
3. Banyaknya ulangan bisa berbeda antar perlakuan
4. Banyaknya perlakuan dan ulangan hanya dibatasi oleh banyaknya unit percobaan yang tersedia

Namun, RAL juga memiliki beberapa kekurangan, antara lain:

1. Informasi karena data yang hilang relatif kecil dibanding rancangan lainnya
2. Jika ada faktor pengganggu, sering kali kurang efisien karena pengacakan tidak dibatasi
3. Jika jumlah ulangan sedikit dan unit percobaan tidak benar-benar homogen, pengulangan akan tidak konsisten

4. Galat percobaan mencakup seluruh keragaman antar unit percobaan kecuali yang disebabkan oleh perlakuan

Kekurangan tersebut menunjukkan kelemahan asumsi unit percobaan relatif homogen di RAL. Sesuai (2) dan (4), galat percobaan menjadi lebih besar jika ada sumber keragaman antar unit percobaan yang tidak disebabkan perlakuan. Walaupun keragaman itu dapat diatasi oleh pengulangan, dibutuhkan beberapa ulangan untuk mengatasi keragaman.

2.2.1 Pengacakan

Penerapan perlakuan pada unit percobaan RAL dilakukan secara acak lengkap terhadap seluruh unit percobaan. Setiap perlakuan mempunyai peluang yang sama besar untuk diterapkan di tiap unit percobaan. Pengacakan dapat dilakukan dengan bantuan sistem lotere/undian, tabel bilangan acak, kalkulator, atau *software* komputer. Tiap pengacakan akan memiliki hasil berbeda. Salah satu cara pengacakan di *software* komputer (misal Excel) dapat dilakukan adalah sebagai berikut:

1. Semua perlakuan disusun secara sistematis (semuanya terurut - r baris pertama adalah r ulangan perlakuan 1, dan seterusnya.)
2. Berikan label nomor 1 sampai n (banyaknya unit percobaan)
3. Bangkitkan bilangan acak sebanyak n, bisa menggunakan fungsi =RAND() di Excel
4. Urutkan perlakuan berdasarkan bilangan acaknya dari terbesar ke terkecil

Pengacakan menggunakan *software* R akan digambarkan melalui studi kasus tersebut:

Suatu Lembaga Penelitian Padi melakukan percobaan untuk mengetahui pengaruh penggunaan berbagai pestisida cair terhadap produksi padi. Percobaan dilakukan pada satu hamparan sawah yang mempunyai tingkat kesuburan atau kandungan bahan organik yang relatif sama. Masing-masing perlakuan diberikan pada lahan seluas 4 x 5 m². Perlakuan yang dicobakan ada 7, yaitu Dol-Mix (1 kg), Dol-Mix (2 kg), DDT + γ -BHC, Azodrin, Dimecron-Boom, Dimecron-Knap, dan tanpa pestisida (sebagai kontrol), masing-masing diulang 4 kali. Respon yang diukur adalah produksi gabah per kg/ha. Hasil dari setiap petakan dikonversi ke dalam kg/ha.

— Mattjik dan Sumertajaya (2002)

Total perlakuan adalah 28. Akan digunakan fungsi `design.crd` dari library `agricolae` di R. Lihat dulu argumen-argumen fungsi tersebut:

```
library(agricolae)
#cek fungsi design.
str(design.crd)
```

```
function (trt, r, serie = 2, seed = 0, kinds = "Super-Duper", randomization = TRUE)
```

Argumen trt akan diisi perlakuan, r akan diisi ulangan. Randomization pasti TRUE karena kita ingin pengacakan dilakukan. Serie merupakan argumen yang berkaitan dengan skema penomoran unit percobaan - jika serie=2, penomoran mulai dari 101. Kita memilih seed=0 agar penomoran berurutan logis (mulai dari 0). Lalu, kita lakukan pengacakan dengan fungsi tersebut:

```
#masukkan perlakuan
perlakuan<-c("DolMix1","DolMix2","DDT","Azodrin","DimecronB","DimecronK","Kontrol")

#buat bagan
baganRAL<-design.crd(trt=perlakuan,r=4,seed=16,serie=0)

#akses output -> hasil design.crd$book
knitr::kable(head(baganRAL$book,n=10))
```

plots	r	perlakuan
1	1	Kontrol
2	1	DimecronB
3	2	Kontrol
4	1	DimecronK
5	2	DimecronK
6	1	DDT
7	2	DimecronB
8	3	DimecronB
9	1	DolMix2
10	2	DolMix2

Hanya ditunjukkan 10 unit percobaan pertama agar tabel tidak terlalu panjang, tetapi jumlah tersebut seharusnya sudah dapat menggambarkan hasil pengacakan bagan percobaan RAL.

2.2.1.1 Pengacakan - edible

Percobaan juga dapat dirancang dengan package edible. Package ini memudahkan pengguna untuk membuat suatu bagan dari suatu konteks (misal, sebuah cerita), dan memudahkan pengguna mengetahui konteks dari suatu rancangan. Untuk install package tersebut, install package simulate terlebih dahulu.

```
remotes::install_github("emitanaka/simulate")
```

Lalu install package edibble, dan install deggust untuk membantu visualiasi bagan percobaan:

```
install.packages("edibble")  
remotes::install_github("emitanaka/deggust")
```

Misal, lihat kembali soal cerita:

Suatu Lembaga Penelitian Padi melakukan percobaan untuk mengetahui pengaruh penggunaan berbagai pestisida cair terhadap produksi padi. Percobaan dilakukan pada satu hamparan sawah yang mempunyai tingkat kesuburan atau kandungan bahan organik yang relatif sama. Masing-masing perlakuan diberikan pada lahan seluas $4 \times 5 \text{ m}^2$. Perlakuan yang dicobakan ada 7, yaitu Dol-Mix (1 kg), Dol-Mix (2 kg), DDT + γ -BHC, Azodrin, Dimecron-Boom, Dimecron-Knap, dan tanpa pestisida (sebagai kontrol), masing-masing diulang 4 kali. Respon yang diukur adalah produksi gabah per kg/ha. Hasil dari setiap petakan dikonversi ke dalam kg/ha.

— Mattjik dan Sumertajaya (2002)

Pembangunan suatu rancangan percobaan akan diawali dengan **design**. Dalam fungsi **design** ada argumen **name** yang berisi nama percobaan yang ingin dilakukan. Oleh karena itu, awali dengan menulis **design(name="nama percobaan")**. Lalu, logika operasi edibble menggunakan operator pipa. Hasil dari operasi sebelumnya dilanjutkan ke operasi setelahnya. Untuk membuat unit percobaan, tuliskan **set_unit**. Ada $7 \cdot 4 = 28$ unit percobaan berupa sawah, sehingga tulis **set_unit(sawah=28)**. Lalu lanjutkan dengan membuat perlakuan, dengan fungsi **set_trts**. Perlakuan merupakan level dari satu taraf, yaitu pestisida. Level tersebut telah dimasukkan ke vektor perlakuan sebelumnya, jadi tulis **set_trts(pestisida=perlakuan)**.

Lalu, alokasikan perlakuan ke unit. Gunakan fungsi **allot_trts(perlakuan ~ unit)** sehingga tuliskan **allot_trts(pestisida~sawah)**. Tentu perlakuan diberikan secara acak sehingga tulis **assign_trts("random")**. Gunakan fungsi **serve_table** untuk melihat bagan percobaan:

```
library(edibble)  
  
desRAL<-design(name="Pengaruh pestisida cair pada produksi padi") %>%  
  set_units(sawah=28) %>%  
  set_trts(pestisida=perlakuan) %>%  
  allot_trts(pestisida ~ sawah) %>%  
  assign_trts("random", seed=420) %>% serve_table
```

```
knitr::kable(head(desRAL,n=10))
```

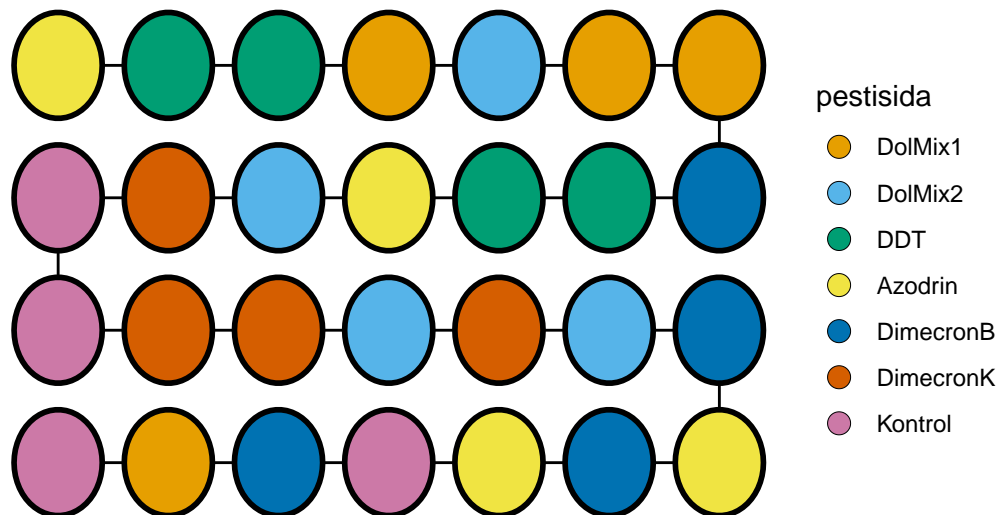
sawah	pestisida
sawah1	Kontrol
sawah2	DolMix1
sawah3	DimecronB
sawah4	Kontrol
sawah5	Azodrin
sawah6	DimecronB
sawah7	Azodrin
sawah8	DimecronB
sawah9	DolMix2
sawah10	DimecronK

Lalu, plot dengan library deggust:

```
deggust::autoplot(desRAL)
```

Pengaruh pestisida cair pada produksi padi

Unit: sawah



Terlihat bahwa tiap perlakuan mendapat 4 unit percobaan. Umumnya, RAL dengan 7 perlakuan dan 28 ulangan dapat dibuat dengan fungsi `takeout`. Ambil `menu_crd`, dengan `t=7`,

n=28. Lalu, lakukan `examine_recipe` untuk melihat cara membangun RAL tersebut

```
crd <- takeout(menu_crd(t = 7, n=28, seed = 420), )  
examine_recipe(crd)
```

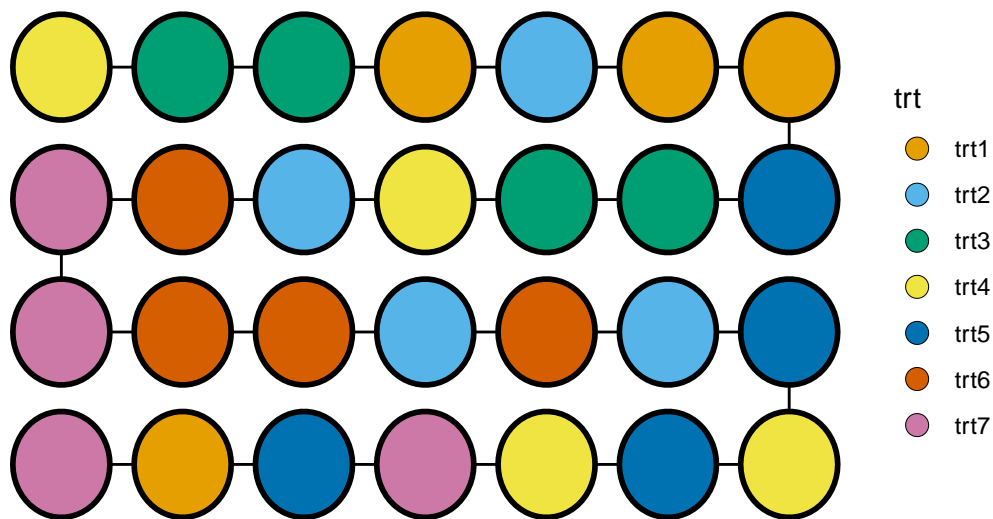
```
design("Completely Randomised Design") %>%  
  set_units(unit = 28) %>%  
  set_trts(trt = 7) %>%  
  allot_trts(trt ~ unit) %>%  
  assign_trts("random", seed = 420) %>%  
  serve_table()
```

Cara pembangunannya sama, begitu juga untuk plot:

```
deggust::autoplot(crd)
```

Completely Randomised Design

Unit: unit



Peneliti juga dapat menuliskan respon yang diinginkan dengan fungsi `set_rcrds`. Perlu dituliskan unit pengamatan respon tersebut, dengan sintaks `set_rcrds(respon = unit)`. Dalam kasus ini tentu diamati hasil produksi dari tiap petakan sawah:

```
desRAL %>% set_rcrds(produksi = sawah)
```

```
# Pengaruh pestisida cair pada produksi padi
# An edibble: 28 x 3
  sawah pestisida produksi
  <unit(28)> <trt(7)> <rcrd>
1   sawah1 Kontrol      o
2   sawah2 DolMix1      o
3   sawah3 DimecronB    o
4   sawah4 Kontrol      o
5   sawah5 Azodrin      o
6   sawah6 DimecronB    o
7   sawah7 Azodrin      o
8   sawah8 DimecronB    o
9   sawah9 DolMix2      o
10  sawah10 DimecronK    o
# ... with 18 more rows
# i Use `print(n = ...)` to see more rows
```

Lalu, record tersebut dapat diberi batasan nilainya. Misal, tidak mungkin produksi negatif, dengan fungsi `expect_rcrds(kondisi respon)`:

```
desRAL %>% set_rcrds(produksi = sawah) %>% expect_rcrds(produksi >= 0)
```

```
# Pengaruh pestisida cair pada produksi padi
# An edibble: 28 x 3
  sawah pestisida produksi
  <unit(28)> <trt(7)> <rcrd>
1   sawah1 Kontrol      o
2   sawah2 DolMix1      o
3   sawah3 DimecronB    o
4   sawah4 Kontrol      o
5   sawah5 Azodrin      o
6   sawah6 DimecronB    o
7   sawah7 Azodrin      o
8   sawah8 DimecronB    o
9   sawah9 DolMix2      o
10  sawah10 DimecronK    o
# ... with 18 more rows
# i Use `print(n = ...)` to see more rows
```

Ini dapat membantu tabulasi data. Jika ada data yang di luar ekspektasi setelah dicatat, unit percobaan tersebut dapat diteliti kembali.

2.2.2 Model

Setelah memberikan perlakuan-perlakuan kepada unit percobaan, dibutuhkan model untuk mengduga pengaruh perlakuan-perlakuan tersebut. Model untuk RAL adalah sebagai berikut:

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij} \begin{cases} i = 1, 2, \dots, t \\ j = 1, 2, \dots, r \end{cases}$$

Dalam model ini, y_{ij} adalah observasi peubah respon di perlakuan (t) ke-i dan ulangan (r) ke-j. Diasumsikan bahwa μ adalah rata-rata umum nilai peubah respon. τ_i adalah efek perlakuan tertentu. μ_i adalah rata-rata peubah respon di amatan yang mendapat perlakuan ke-i. Di model tetap, dapat disusun rata-rata perlakuan μ_i dari μ dan τ_i :

$$\mu_i = \mu + \tau_i$$

Dan, anggap bahwa:

$$\frac{\sum_{i=1}^t \mu_i}{t} = \mu$$

Sebagai konsekuensi dari asumsi tersebut,

$$\begin{aligned} \frac{\sum_{i=1}^t \mu_i}{t} &= \frac{\sum_{i=1}^t (\mu + \tau_i)}{t} \\ \mu &= \frac{t\mu}{t} + \frac{\sum_{i=1}^t \tau_i}{t} = \mu + \frac{\sum_{i=1}^t \tau_i}{t} \\ \frac{\sum_{i=1}^t \tau_i}{t} &= 0 \end{aligned}$$

Yang berarti $\sum_{i=1}^t \tau_i = 0$. Maka dari itu, asumsi bahwa jumlah dari rata-rata perlakuan dibagi jumlah perlakuan adalah rata-rata keseluruhan memiliki implikasi bahwa jumlah efek tiap perlakuan nol. Asumsi tersebut tidak akan benar jika jumlah ulangan tiap perlakuan tidak sama¹. Oleh karena itu, **asumsi bahwa $\sum_{i=1}^t \tau_i = 0$** yang dituliskan di beberapa teks, seperti Mattjik dan Sumertajaya (2002), sebenarnya **sama dengan asumsi bahwa semua perlakuan mendapat ulangan sama**.

¹Anggap ada tiga perlakuan dengan rata-rata 1, 2, 3, masing-masing diulang 2,3, dan 3 kali. Rata-rata umum adalah $\frac{1+1+2+2+2+3+3+3}{8} = \frac{2+6+9}{8} = \frac{17}{8}$. Namun, jumlah rata-rata perlakuan dibagi jumlah perlakuan adalah $\frac{6}{3} = 2$.

Suplemen untuk Montgomery (2017) mengatakan bahwa dalam kasus umum, rata-rata keseluruhan dapat didefinisikan sebagai $\sum_{i=1}^t w_i \mu_i = \mu$, dengan $\sum_{i=1}^t w_i = 1$. Dalam kata lain, jumlah terboboti dari rata-rata kelompok adalah rata-rata keseluruhan. Definisi tersebut memberi implikasi bahwa $\sum_{i=1}^t w_i \tau_i = 0$, yang diturunkan dengan prosedur sama seperti sebelumnya. Dalam kasus ulangan sama, bobot tiap perlakuan sama, yaitu $\frac{1}{t}$ ². Secara umum, jika N adalah total unit percobaan dan r_i ulangan perlakuan ke- i , bobot memiliki formulasi $w_i = \frac{r_i}{N}$ (ini seharusnya intuitif karena kita memboboti sesuai proporsi perlakuan tersebut dari seluruh unit percobaan).

Asumsi yang analog untuk model acak adalah $\tau_i \sim N(0, \sigma^2 \tau)$.

ε_{ij} adalah komponen galat acak yang merupakan semua sumber keragaman lain. Sumber tersebut, misal, adalah pengukuran, perbedaan antara satuan percobaan, efek dari lingkungan, dan lain-lain. Galat tersebut diasumsikan memiliki nilai harapan nol. Secara matematis:

$$E(\varepsilon_{ij}) = 0$$

2.2.3 Statistik

Setelah melihat model, ada beberapa statistik dari peubah respon yang akan digunakan dalam ANOVA. $y_{i.}$ adalah total nilai peubah respon dari seluruh unit percobaan dengan perlakuan ke- i . Dalam kata lain, di perlakuan ke- i , nilai peubah respon dari r ulangan ditambah. \bar{y}_i adalah rata-rata nilai peubah respon dari seluruh observasi dengan perlakuan ke- i . Sedangkan, $y_{..}$ adalah total nilai peubah respon dari seluruh observasi, dengan rata-ratanya dinotasikan $\bar{y}_{..}$. Rumus-rumus statistik tersebut adalah:

$$y_{i.} = \sum_{j=1}^r y_{ij} \quad \bar{y}_i = \frac{y_{i.}}{r} \quad i = 1, 2, \dots, t$$

$$y_{..} = \sum_{i=1}^t \sum_{j=1}^r y_{ij} \quad \bar{y}_{..} = \frac{y_{..}}{tr}$$

Kita dapat mengatakan bahwa $tr = N$, atau jumlah total unit percobaan.

²Jika kita masukkan bobot tersebut, $\frac{\sum_{i=1}^t \mu_i}{t} = \mu$. Ini berarti jika ulangan sama asumsi jumlah efek perlakuan nol terpenuhi

2.2.4 Dekomposisi Jumlah Kuadrat

ANOVA adalah singkatan dari *analysis of variance*. Analisis ini berkaitan dengan keragaman karena tujuannya adalah mempartisi keragaman menjadi beberapa bagian. Ukuran keragaman total adalah **jumlah kuadrat total** (JK_T), dengan rumus:

$$\begin{aligned} JK_T &= \sum_{i=1}^t \sum_{j=1}^r (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^t \sum_{j=1}^r (y_{ij} - \bar{y}_{i.})^2 + r \sum_{i=1}^t (\bar{y}_{i.} - \bar{y}_{..})^2 \\ &= JK_G + JK_P \end{aligned}$$

Jumlah kuadrat tersebut dapat dihitung menggunakan:

$$\begin{aligned} FK &= \frac{y_{..}^2}{tr} \\ JK_T &= \sum_{i=1}^t \sum_{j=1}^r (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^t \sum_{j=1}^r y_{ij}^2 - FK \\ JK_P &= r \sum_{i=1}^t (\bar{y}_{i.} - \bar{y}_{..})^2 = \sum_{i=1}^t \frac{y_{i.}^2}{r} - FK \\ JK_G &= JK_T - JK_P \end{aligned}$$

Jika percobaan memiliki ulangan sama. Jika percobaan memiliki ulangan tak sama, jumlah ulangan r harus diubah menjadi r_i , jumlah ulangan perlakuan ke- i . Oleh karena penyesuaian tersebut, rumus-rumus jumlah kuadrat sebagai berikut:

$$\begin{aligned} FK &= \frac{y_{..}^2}{\sum_{i=1}^t r_i} \\ JK_T &= \sum_{i=1}^t \sum_{j=1}^{r_i} (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^t \sum_{j=1}^{r_i} y_{ij}^2 - FK \\ JK_P &= \sum_{i=1}^t \sum_{j=1}^{r_i} (\bar{y}_{i.} - \bar{y}_{..})^2 = \sum_{i=1}^t \frac{y_{i.}^2}{r_i} - FK \\ JK_G &= JK_T - JK_P \end{aligned}$$

2.2.5 Asumsi Galat

Dari jumlah kuadrat, dapat ditemukan kuadrat tengah. Kuadrat tengah adalah jumlah kuadrat dibagi derajat bebasnya. Kuadrat tengah galat merupakan penduga bagi ragam σ^2 , di mana $E(KT_G) = \sigma^2$ dan jika semua $\tau_i = 0$ kuadrat tengah perlakuan juga merupakan

penduga ragam (nilai harapannya akan sama dengan σ^2). Hasil ini didapat dari dua asumsi. Asumsi pertama adalah mengenai ragam galat:

$$Var(\varepsilon_{ij}) = E[\{\varepsilon_{ij} - E(\varepsilon_{ij})\}^2] = E(\varepsilon_{ij}^2) = \sigma^2$$

Karena nilai harapan $E(\varepsilon_{ij})$ sama dengan 0. Selain itu, terdapat asumsi mengenai kebebasan galat:

$$\begin{aligned} Cov(\varepsilon_{ij}\varepsilon_{ik}) &= E[(\varepsilon_{ij} - E[\varepsilon_{ij}])(\varepsilon_{ik} - E[\varepsilon_{ik}])] \\ &= E(\varepsilon_{ij}\varepsilon_{ik}) = 0 \end{aligned}$$

2.2.6 Hipotesis dan ujinya

Dinyatakan bahwa $E(KT_P) = E(KT_G)$ saat perlakuan tidak memiliki efek. Secara formal, ini dinyatakan sebagai:

$$\begin{aligned} H_0 &= \tau_1 = \dots = \tau_t = 0 \\ H_1 &= \text{Setidaknya ada satu } i \text{ dengan } \tau_i \neq 0 \end{aligned}$$

Ini sama dengan menyatakan:

$$\begin{aligned} H_0 &= \mu_1 = \dots = \mu_t = \mu \\ H_1 &= \text{Setidaknya ada satu } i \text{ dengan } \mu_i \neq \mu \end{aligned}$$

Uji hipotesis ini dilakukan dengan uji F. Ini dikarenakan JK_G/σ^2 dan JK_P/σ^2 menyebar chi-kuadrat secara bebas. Jika dilakukan pembagian dua peubah chi-kuadrat saling bebas, dibagi dengan rasio derajat bebas kedua peubah tersebut, akan muncul peubah yang menyebar F. σ^2 akan hilang karena saling terbagi. Selain itu, kuadrat tengah merupakan jumlah kuadrat dibagi derajat bebas sehingga KT_G dan KT_P menyebar F.

2.2.7 Review MKT

MKT pada dasarnya adalah mencari nilai penduga yang meminumkan jumlah kuadrat galat $\sum_{i=1}^t \sum_{j=1}^r \hat{\varepsilon}_{ij}^2 = \sum_{i=1}^t \sum_{j=1}^r (y_{ij} - \hat{\mu} - \hat{\tau}_i)^2$. Diketahui dari Kalkulus bahwa mencari nilai ekstrim suatu fungsi (minimum atau maksimum) dilakukan dengan mencari nilai yang menghasilkan turunan pertama nol. Karena ada banyak rumus, akan digunakan turunan parsial:

$$\begin{aligned}\frac{\delta L}{\delta \mu} &= 0 \\ \frac{\delta L}{\delta \tau_i} &= 0, i = 1, 2, \dots, t\end{aligned}$$

Penurunan, dan asumsi bahwa $\sum_{i=1}^t \tau_i = 0$ akan menghasilkan:

$$tr\hat{\mu} + r \left(\sum_{i=1}^t \hat{\tau}_i \right) = tr\hat{\mu} = y_{..}$$

Oleh karena itu, penduga bagi μ , $\hat{\mu}$ adalah $y_{..}/tr = \bar{y}_{..}$. Selain itu, dapat ditemukan bahwa:

$$\begin{aligned}r\bar{y}_{..} + r\hat{\tau}_i &= y_{i.} \\ r\hat{\tau}_i &= y_{i.} - r\bar{y}_{..} \\ \hat{\tau}_i &= \bar{y}_{i.} - \bar{y}_{..}\end{aligned}$$

Oleh karena itu, kita telah menemukan penduga-penduga tersebut. Terakhir, ternyata hasil penurunan akan mengharuskan:

$$\sum_{i=1}^t \sum_{j=1}^r (y_{ij} - \mu - \tau_i) = 0$$

Ini berarti rata-rata penduga galat juga akan 0, yang memenuhi salah satu asumsi ANOVA. Sebagai tambahan, dalam model acak penduga bagi ragam pengaruh perlakuan adalah:

$$\hat{\sigma}_{\tau}^2 = \frac{KTP - KTG}{r}$$

2.2.8 Review Asumsi-Asumsi

Dapat disimpulkan bahwa asumsi-asumsi yang dibutuhkan dalam penurunan uji F adalah:

1. $E(\varepsilon_{ij}) = E(\varepsilon_{i.}) = E(\varepsilon_{..}) = 0$ yang terpenuhi melalui metode MKT.
2. $\sum_{i=1}^t \tau_i = 0$, yang terpenuhi dari jumlah ulangan sama di satu perlakuan, serta digunakan sebagai batasan dalam MKT. Ada bentuk umum jika asumsi tersebut tidak terpenuhi.
3. $E(\varepsilon_{ij}^2) = \sigma^2$ yang dibutuhkan untuk memastikan kuadrat tengah dapat menduga ragam secara tak bias.
4. $Cov(\varepsilon_{ij}\varepsilon_{kl}) = 0$, atau galat saling bebas, juga untuk memastikan ketidakhbiasan.
5. $\varepsilon_{ij} \sim N(0, \sigma^2)$ agar dapat dilakukan Uji-F.

2.2.9 Tabel ANOVA

Uji-F dilakukan dengan membangun tabel ANOVA. Tabel ANOVA untuk RAL ulangan sama sebagai berikut:

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
Perlakuan	t-1	JKP	JKP/dbP	KTP/KTG	
Galat	t(r-1)	JKG	JKG/dbG		
Total	tr-1	JKT			

Dan untuk ulangan tak sama derajat bebas galat menjadi $\sum_{i=1}^t (r - i - 1)$ dan derajat bebas total $\sum_{i=1}^t r_i$.

2.2.9.1 Aplikasi: Lengkapi Tabel ANOVA!

Terdapat sebuah percobaan ulangan sama dengan empat perlakuan dan tiga ulangan. Lengkapi tabel ANOVA tersebut, jika diketahui JK_P dan JK_T !

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
Perlakuan	t-1	120	JKP/dbP	KTP/KTG	
Galat	t(r-1)	JKG	JKG/dbG		
Total	tr-1	170			

Untuk ulangan tak sama, tuliskan tabel ANOVA jika percobaan tersebut memiliki 3 ulangan pada perlakuan pertama dan terakhir, dan 2 ulangan pada perlakuan kedua, serta 4 ulangan pada perlakuan ketiga! Jumlah JK_P dan JK_T tetap sama.

2.2.10 Praktek menggunakan R

Setelah membuat bagan percobaan, kita akan melakukan ANOVA. Kasus yang dipakai akan berbeda, yaitu:

Karantina Tumbuhan ingin mengetahui pengaruh fumigan *Methyl Bromide* (CH_3Br) sebagai pembasmi serangga gudang terhadap daya tumbuh benih kacang hijau. Untuk itu, dilakukan percobaan sebagai berikut: benih kacang hijau diberi fumigan dengan dosis 0 (kontrol), 16 gr/m^2 , 32 gr/m^2 , 48 gr/m^2 , 64 gr/m^2 . Fumigasi dilakukan selama 2 jam. Benih kacang hijau yang sudah

difumigasi dikecambahkan dengan metode kertas hisp (*blotter test*). Benih yang dikecambahkan diasumsikan homogen. Setelah 7 hari diperoleh hasil [sebagai berikut](#).

Analisis dimulai dari mengambil data tersebut:

```
library(google sheets4)
#baca sheet

DataRAL<-read_sheet("https://docs.google.com/spreadsheets/d/1Bzm_R2Zd4Zbij7B07LGDJW83DU6T3")
```

! Using an auto-discovered, cached token.

To suppress this message, modify your code or options to clearly consent to the use of a cached token.

See gargle's "Non-interactive auth" vignette for more details:

<<https://gargle.r-lib.org/articles/non-interactive-auth.html>>

i The google sheets4 package is using a cached token for 'muh.ammarsahab@apps.ipb.ac.id'.

v Reading from "DataRAL".

v Range 'Sheet1'.

```
knitr::kable(DataRAL)
```

Dosis	1	2	3	4	5	6	7	8
0	100	100	100	100	100	100	100	100
16	100	100	100	100	100	100	100	100
32	90	88	92	94	90	88	86	94
48	80	80	82	78	84	76	82	78
64	90	80	92	78	82	88	94	76

Akan dilakukan beberapa modifikasi terhadap data tersebut. Pertama, data ditransformasi menjadi bentuk *long*. Ini dikarenakan tabulasi data yang mencatat tiap ulangan dalam satu kolom kurang cocok untuk analisis. Tabulasi tersebut akan diubah sehingga data peubah respon semua ulangan dan perlakuan berada di satu kolom:

```
library(reshape2)

RALMelt<- melt(DataRAL,
               #variabel yang membedakan tiap baris di tabulasi asli:
               id.vars=c("Dosis"),
               #kolom yang ingin digabung jadi 1:
               measured.vars=as.character(seq(1,8)),
               value.name="Perkecambahan")

colnames(RALMelt)[2]<-"Ulangan"
knitr::kable(head(RALMelt,n=10))
```

Dosis	Ulangan	Perkecambahan
0	1	100
16	1	100
32	1	90
48	1	80
64	1	90
0	2	100
16	2	100
32	2	88
48	2	80
64	2	80

Tanpa mengubah nama kolom, nama kolom Ulangan akan secara default menjadi “variable”. Jika opsi value.name tidak diberi di fungsi melt, kolom Perkecambahan akan menjadi “value”. Lalu, kita buat dosis menjadi sebuah faktor:

```
RALMelt$Dosis<-as.factor(RALMelt$Dosis)
```

Setelah itu, dapat langsung dilakukan ANOVA.

```
aov_RAL <- aov(Perkecambahan ~ Dosis, data = RALMelt)
summary(aov_RAL)
```

```

          Df Sum Sq Mean Sq F value    Pr(>F)
Dosis      4 2556.4   639.1    51.36 3.6e-14 ***
Residuals 35  435.5    12.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil ANOVA yang lebih rapih dapat ditunjukkan dengan `broom::tidy`. Pada intinya, fungsi ini mengubah output anova menjadi suatu *tibble*, mirip `data.frame`, sehingga dapat dibuat tabel:

```
broom::tidy(aov_RAL) |> knitr::kable()
```

term	df	sumsq	meansq	statistic	p.value
Dosis	4	2556.4	639.10000	51.3628	0
Residuals	35	435.5	12.44286	NA	NA

P-value statistik-F tersebut sangat kecil. Ini berarti kemungkinan menemukan nilai F sama atau lebih dari statistik F yang kita temukan sangat kecil, jika H_0 benar dan tidak ada efek perlakuan. Oleh karena itu, memakai α 0.001 dan selebihnya (jika dianggap bahwa kemungkinan sekecil tersebut sudah cukup untuk menolak H_0), disimpulkan bahwa perlakuan memiliki efek pada respon.

2.2.11 Analisis Percobaan di Python

Selain menggunakan R dan secara manual, software *open-source* lain yang dapat digunakan untuk Perancangan Percobaan adalah Python. Kita akan menggunakan modul `pandas` dan `doe` dari Python. Modul `doe` memerlukan tabulasi data dengan bentuk berbeda dari R. Tiap kolom harus mengandung data tiap perlakuan. Ini merupakan kebalikan dari tabulasi asli, yaitu tiap baris mengandung data perlakuan dan tiap kolom mengandung data ulangan.

Oleh karena itu, kita mulai dengan membaca data melalui `pd.read_csv()`. Di python, ubah saja tautan google sheets <https://docs.google.com/spreadsheets/d/tautan/edit?usp=sharing> menjadi <https://docs.google.com/spreadsheets/d/tautan/export?gid=0&format=csv>. Bagian terakhir setelah tautan diubah menjadi `export?gid=0&format=csv`,

```
import pandas as pd
```

```
RALData=pd.read_csv("https://docs.google.com/spreadsheets/d/1Bzm_R2Zd4Zbij7B07LGDJW83DU6T3RALData")
```

	Dosis	1	2	3	4	5	6	7	8
0	0	100	100	100	100	100	100	100	100
1	16	100	100	100	100	100	100	100	100
2	32	90	88	92	94	90	88	86	94
3	48	80	80	82	78	84	76	82	78
4	64	90	80	92	78	82	88	94	76

Reshaping dapat dilakukan sehingga sama seperti tabulasi data R. Agar reshaping tersebut dapat dilakukan, perlu dibuat suatu list yang angka dari satu sampai delapan untuk menandakan ulangan. Anggota list tersebut harus berbentuk karakter. Oleh karena itu, fungsi `range(1,9)` digunakan untuk membuat suatu deret angka dari 1 sampai $9 - 1 = 8$. Lalu, numerik-numerik tersebut diubah menjadi karakter dan dimasukkan ke dalam list.

Fungsi untuk mengubah numerik ke karakter adalah `str`. Cara mengaplikasikan fungsi tersebut ke sejumlah suatu deret hasil fungsi `range` atau list pada umumnya adalah `map(fungsi, range)`, atau `fungsi(x) for x in range`. Lalu, `list()` mengubah objek menjadi list, atau dapat juga dilakukan dengan `[isi list]`. Hasil kedua kode ini sama:

```
list(map(str,range(1,9)))
```

```
['1', '2', '3', '4', '5', '6', '7', '8']
```

```
[str(x) for x in range(1,9)]
```

```
['1', '2', '3', '4', '5', '6', '7', '8']
```

Lalu melt data tersebut:

```
RALMeltpy=pd.melt(RALData, id_vars=['Dosis'], value_vars=[str(x) for x in range(1,9)])
```

```
RALMeltpy.head()
```

	Dosis	variable	value
0	0	1	100
1	16	1	100
2	32	1	90
3	48	1	80
4	64	1	90

Nama kolom pertama dan kedua perlu diubah, menggunakan fungsi `rename`. Fungsi tersebut diaplikasikan ke suatu dataset dengan `dataset.rename()`. Lalu kolom-kolom yang diubah namanya dapat dispesifikasi dengan `columns={'kolomlama:kolombaru', 'kolomlama:kolombaru'}`. `inplace=True` akan mengubah dataset secara langsung, sedangkan `inplace=False` akan membuat suatu objek baru:

```
RALMeltpy.rename(columns = {'variable':'Ulangan', 'value':'Produksi'}, inplace = True)
```

```
RALMeltpy.head()
```

	Dosis	Ulangan	Produksi
0	0	1	100
1	16	1	100
2	32	1	90
3	48	1	80
4	64	1	90

Lalu gunakan statsmodels untuk ANOVA. OLS digunakan untuk membangun suatu model linear. Sintaksnya relatif mirip dengan R, yaitu `ols('respon ~ peubah')` dengan `C(Peubah)` menandakan suatu peubah kategorik. Hasil model tersebut diakses dengan `model.fit()`:

```
import statsmodels.api as sm
```

```
C:\Users\Acer\AppData\Local\R\win-library\4.2\reticulate\python\rpytools\loader.py:39: FutureWarning:
  module = _import(
C:\Users\Acer\AppData\Local\Programs\Python\PYTHON~2\lib\site-packages\statsmodels\compat\pandas.py:10:
  from pandas import Int64Index as NumericIndex
```

```
from statsmodels.formula.api import ols
```

```
RALlm= ols('Produksi ~ C(Dosis)',data=RALMeltpy)
fitRAL=RALlm.fit()
```

Lalu, lakukan ANOVA dari objek fit tersebut:

```
table = sm.stats.anova_lm(fitRAL)
print(table)
```

	df	sum_sq	mean_sq	F	PR(>F)
C(Dosis)	4.0	2556.4	639.100000	51.362801	3.597935e-14
Residual	35.0	435.5	12.442857	NaN	NaN

Hasil sama dengan R.

2.3 RAKL

Rancangan Acak Kelompok Lengkap (RAKL) dapat juga disebut sebagai Rancangan Kelompok Teracak Lengkap (RKTL) atau *Randomized Complete Block Design* (RCBD). Karakteristik RAKL adalah:

1. Kondisi unit percobaan tidak homogen. Keheterogenan unit percobaan berasal dari satu sumber keragaman. Ketidakhomogenan dapat dikendalikan dengan pengelompokan sehingga RAKL memungkinkan pengendalian ragam satu arah.
2. Faktor di luar perlakuan dan kelompok relatif homogen.
3. Keragaman antarkelompok tinggi, tetapi di dalam kelompok kecil.

Oleh karena karakteristik pertama, RAKL cocok digunakan di lapangan, yang memiliki perubahan kondisi (seperti kelembaban, suhu, dsb). Karena RAKL masih dalam satu faktor, perlakuan merupakan level-level dari satu faktor tertentu.

Kelebihan dalam menggunakan RAKL adalah sebagai berikut:

1. Analisis statistik yang relatif sederhana bahkan dengan data hilang.
2. Galat percobaan akan berkurang sehingga perbandingan antar perlakuan meningkat, dengan kata lain akan memberikan hasil yang lebih akurat daripada RAL karena pengelompokan.
3. Mengizinkan perhitungan kesalahan yang tidak bias untuk perlakuan tertentu.
4. Fleksibilitas penuh, bisa memiliki sejumlah perlakuan dan kelompok.

Kekurangan dalam menggunakan RAKL adalah sebagai berikut.

1. Tidak cocok untuk jumlah perlakuan besar karena kelompok jadi semakin besar.
2. Interaksi antara kelompok dan perlakuan meningkatkan kesalahan **di model tetap**.
3. Ragam besar antara unit percobaan dalam kelompok dapat menghasilkan galat yang besar. Jika ada data hilang, RAKL mungkin kurang efisien daripada RAL.

Poin kedua relatif penting - kita ingin perlakuan dan kelompok tidak memengaruhi satu sama lain. Efek semua perlakuan tetap sama di semua kelompok, dan sebaliknya. Namun, asumsi tanpa interaksi ini dapat dilanggar jika menggunakan model di mana kelompok dipilih secara acak.

2.3.1 Pengacakan

Pengacakan RAKL dilakukan di tiap kelompok. Algoritme pengacakan RAKL adalah sebagai berikut:

1. Buat pemisalan nomor perlakuan $1, 2, \dots, t$.
2. Beri nomor tersebut di setiap kelompok $1, 2, \dots, t$.

3. Secara acak mengalokasikan perlakuan t ke unit percobaan di setiap kelompok. Pengacakan dilakukan pada masing-masing kelompok dan setiap perlakuan hanya muncul sekali dalam setiap kelompok.

Implementasi di R memakai library agricolae akan menggunakan contoh berikut:

Percobaan lain pada Balai Penelitian Padi tersebut adalah ingin mengetahui pengaruh kerapatan benih terhadap produksi padi (kg/ha) pada varietas IR8. Kerapatan benih yang dicobakan adalah 25 kg benih/ha, 50 kg benih/ha, 75 kg benih/ha, 100 kg benih/ha, 125 kg benih/ha, 150 kg benih/ha. Lahan yang digunakan di perbukitan (lereng bukit), sehingga tingkat kesuburan tidak sama. Untuk itu petak dibagi jadi 4 kelompok. Masing-masing benih ditanam pada lahan seluas 5 x 5 m² serta hasilnya dikonversi ke satuan kg/ha.

Ada enam perlakuan, dan empat kelompok. Akan ada 24 unit percobaan.

```
library(agricolae)
str(design.rcbd)
```

```
function (trt, r, serie = 2, seed = 0, kinds = "Super-Duper", first = TRUE,
         continue = FALSE, randomization = TRUE)
```

Argumen r dalam kasus ini akan menjadi jumlah kelompok. Maka dari itu, kita dapat lakukan pembuatan bagan:

```
perlakuanRAK<-c(as.character(seq(25,150,25)))
perlakuanRAK<-paste(perlakuanRAK, "kg/ha", sep=" ")
baganRAK<-design.rcbd(perlakuanRAK,4,seed=78)
knitr::kable(baganRAK$sketch)
```

50 kg/ha	75 kg/ha	100 kg/ha	150 kg/ha	25 kg/ha	125 kg/ha
75 kg/ha	50 kg/ha	150 kg/ha	25 kg/ha	125 kg/ha	100 kg/ha
75 kg/ha	125 kg/ha	25 kg/ha	100 kg/ha	50 kg/ha	150 kg/ha
25 kg/ha	75 kg/ha	50 kg/ha	150 kg/ha	125 kg/ha	100 kg/ha

Perhatikan bahwa satu perlakuan hanya muncul sekali di tiap baris, sesuai konsep pengacakan RAKL yang dilakukan di dalam tiap baris.

2.3.1.1 Pengacakan edible

Dalam RAK, ada satuan kelompok. Di tiap kelompok, terdapat 6 petak sesuai dengan perlakuan. Maka dari itu, masukkan kelompok dan petak di `set_units`. Gunakan `nested_in(blok, jml unit)` untuk menandakan bahwa petak di dalam kelompok. Selebihnya sama saja. Kerapatan dialokasikan ke petak secara acak:

```
library(edibble)

desRAKL<-design(name="Pengaruh kerapatan benih pada produksi padi") %>%
  set_units(kelompok=4,
            petak=nested_in(kelompok, 6)) %>%
  set_trts(kerapatan=perlakuanRAK) %>%
  allot_trts(kerapatan ~ petak) %>%
  assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desRAKL,n=10))
```

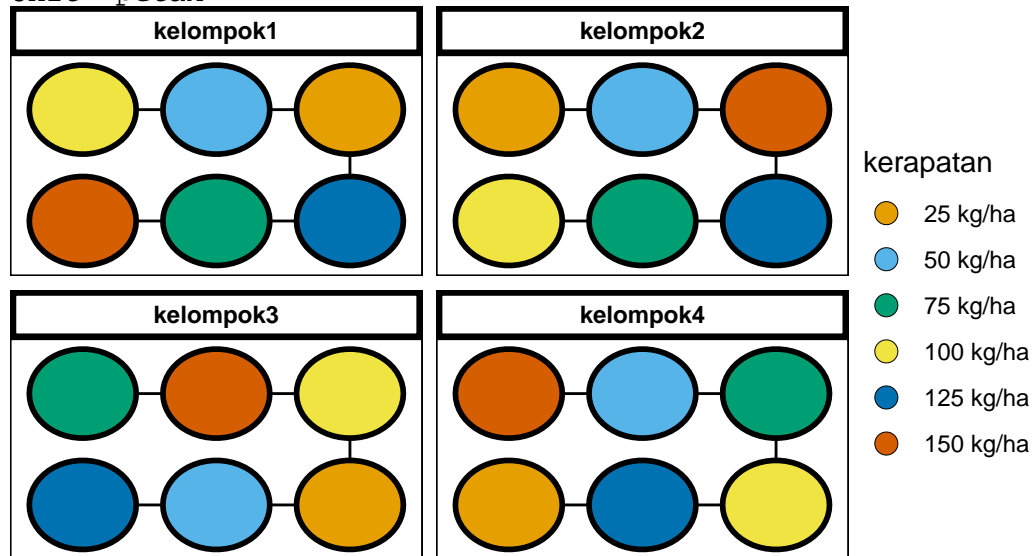
kelompok	petak	kerapatan
kelompok1	petak1	150 kg/ha
kelompok1	petak2	75 kg/ha
kelompok1	petak3	125 kg/ha
kelompok1	petak4	25 kg/ha
kelompok1	petak5	50 kg/ha
kelompok1	petak6	100 kg/ha
kelompok2	petak7	100 kg/ha
kelompok2	petak8	75 kg/ha
kelompok2	petak9	125 kg/ha
kelompok2	petak10	150 kg/ha

Plot rancangan tersebut:

```
deggust::autoplot(desRAKL)
```

Pengaruh kerapatan benih pada produksi padi

Unit: petak



Rancangan yang dibuat dapat dibandingkan dengan default RCBD dari edibble:

```
rcbd <- takeout(menu_rcbd(t = 6, r = 4, seed=420))
examine_recipe(rcbd)
```

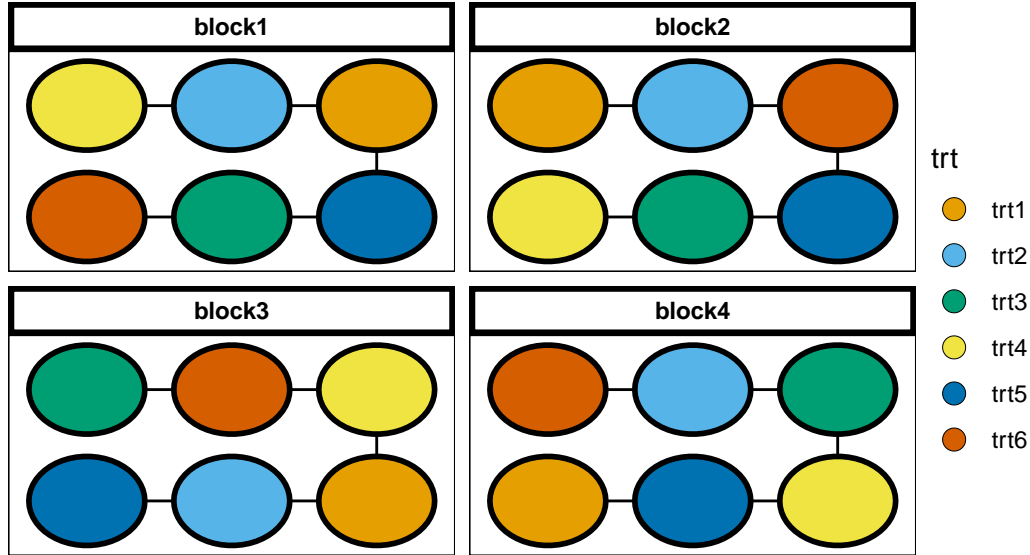
```
design("Randomised Complete Block Design") %>%
  set_units(block = 4,
            unit = nested_in(block, 6)) %>%
  set_trts(trt = 6) %>%
  allot_trts(trt ~ unit) %>%
  assign_trts("random", seed = 420) %>%
  serve_table()
```

Kode yang dibuat relatif mirip. Ada unit yang *nested in* blok, lalu perlakuan diberikan ke unit. Plot juga akan sama:

```
deggust::autoplot(rcbd)
```

Randomised Complete Block Design

Unit: unit



2.3.2 Model Linear Aditif

$$y_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij}$$

Model linear aditif tersebut relatif sama dengan RAL. y_{ij} merupakan nilai peubah respon di perlakuan ke- i dan kelompok ke- j , τ_i merupakan efek pengaruh ke- i , dengan $\sum_{i=1}^t \tau_i = 0$, dan $\varepsilon_{ij} \sim N(0, \sigma^2)$ adalah pengaruh acak di perlakuan ke- i dan kelompok ke- j . Aspek yang berbeda di model tersebut adalah β_j , atau efek dari kelompok ke- j . Sama seperti efek perlakuan, diasumsikan $\sum_{j=1}^r \beta_j = 0$ untuk model tetap dan $\beta_j \sim N(0, \sigma_\beta^2)$.

Oleh karena itu, ada dua hipotesis. Hipotesis pertama, seperti RAL, adalah mengenai **pengaruh perlakuan**:

$$H_0 : \tau_1 = \dots = \tau_t = 0$$

$$H_1 : \text{Setidaknya ada satu } i \text{ di mana } \tau_i \neq 0$$

Yaitu, bahwa semua perlakuan tidak berpengaruh atau sebaliknya, terdapat sedikitnya satu perlakuan yang berpengaruh. Selain itu, ada hipotesis mengenai **pengaruh kelompok**:

$$H_0 : \beta_1 = \dots = \beta_r = 0$$

$$H_1 : \text{Setidaknya ada satu } j \text{ di mana } \beta_j \neq 0$$

Uji hipotesis kelompok tidak selalu dapat dilakukan. Montgomery (2017) mengatakan bahwa jika asumsi normalitas tidak dipenuhi, uji-F bukan cara yang baik untuk menguji efek pengelompokan. Namun, format uji tersebut masih dapat dilakukan sebagai perkiraan. KT_B/KT_G belum tentu dapat dibandingkan dengan sebaran tertentu, tetapi masih dapat dilihat besarnya secara *common sense*.

Dari MKT, penduga bagi β_j adalah $\bar{y}_{j.} - \bar{y}_{..}$. Ini dikarenakan:

$$\begin{aligned}\mu_j &= \mu + \beta_j \\ \hat{\mu}_j &= \bar{y}_{j.} = \bar{y}_{..} + \hat{\beta}_j\end{aligned}$$

2.3.3 ANOVA

Jumlah kuadrat yang dihitung sama dengan RAL, dengan tambahan JK_B (baris/kelompok) yang memiliki rumus $\sum_{j=1}^r (\bar{y}_{.j} - \bar{y}_{..})^2$. Oleh karena itu, di RAKL dihitung:

$$\begin{aligned}FK &= \frac{y_{..}^2}{tr} \\ JK_T &= \sum_{i=1}^t \sum_{j=1}^r (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^t \sum_{j=1}^r y_{ij}^2 - FK \\ JK_P &= r \sum_{i=1}^t (\bar{y}_{i.} - \bar{y}_{..})^2 = \sum_{i=1}^t \frac{y_{i.}^2}{r} - FK \\ JK_B &= t \sum_{j=1}^r (\bar{y}_{.j} - \bar{y}_{..})^2 = \sum_{j=1}^r \frac{y_{.j}^2}{t} - FK \\ JK_G &= JK_T - JK_P - JK_B\end{aligned}$$

Sehingga konstruksi tabel ANOVA sebagai berikut:

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
Perlakuan	t-1	JKP	JKP/dbP	KTP/KTG	
Blok	r-1	JKB	JKB/dbB	KTB/KTG	
Galat	(t-1)(r-1)	JKG	JKG/dbG		
Total	tr-1	JKT			

Jumlah derajat bebas perlakuan, blok, dan total intuitif - mereka berturut-turut merupakan jumlah perlakuan dikurangi satu, blok dikurangi satu, dan unit percobaan dikurangi satu. Derajat bebas galat ditemukan dengan $tr-1-(t-1)-(r-1) = tr-t-r+1 = (t-1)(r-1)$.

2.3.4 Analisis dengan R dan Python

Untuk menunjukkan analisis data RAKL di R, akan dipakai studi kasus ini:

Untuk melihat keefektifan pengaruh pemupukan terhadap produksi suatu varietas padi dilakuakn percobaan di rumah kaca sebagai berikut: 9 kombinasi perlakuan yang dicobakan dengan pupuk K dan P dengan komposisi 2:1 (K2P1), 2:2 (K2P2), ..., 3:4 (K3P4) ditambah sebuah kontrol (K0P0). Setiap perlakuan diulang dalam 3 blok (timur, tengah, barat). Data pengamatannya diperoleh [sebagai berikut](#)

Untungnya, data memiliki format yang mudah untuk dianalisis. Oleh karena itu, load data tersebut:

```
library(google Sheets4)
DataRAKL<-read_sheet("https://docs.google.com/spreadsheets/d/1uFi_Njharot8G8-DvHWPhNLLqJ72
```

```
v Reading from "RAKL_P2 Rancob".
```

```
v Range 'Sheet1'.
```

```
DataRAKL$Produksi<-as.numeric(DataRAKL$Produksi)

#pastikan perlakuan dan kelompok berbentuk faktor
DataRAKL$Kelompok<-as.factor(DataRAKL$Kelompok)
DataRAKL$Perlakuan<-as.factor(DataRAKL$Perlakuan)
knitr::kable(DataRAKL)
```

Perlakuan	Kelompok	Produksi
Kontrol	1	10.19
Kontrol	2	9.26
Kontrol	3	12.73
K2P1	1	32.02
K2P1	2	25.76
K2P1	3	19.72
K2P2	1	23.91
K2P2	2	21.99
K2P2	3	21.42
K2P3	1	17.15
K2P3	2	15.66
K2P3	3	10.37

Perlakuan	Kelompok	Produksi
K2P4	1	10.35
K2P4	2	13.31
K2P4	3	14.31
K3P1	1	21.98
K3P1	2	19.43
K3P1	3	16.16
K3P2	1	18.08
K3P2	2	13.50
K3P2	3	18.32
K3P3	1	18.07
K3P3	2	14.01
K3P3	3	14.39
K3P4	1	12.37
K3P4	2	16.32
K3P4	3	10.20

Dan langsung lakukan analisis:

```
modelRAKL<-aov(Produksi~Perlakuan+Kelompok,DataRAKL)
summary(modelRAKL)
```

```

              Df Sum Sq Mean Sq F value    Pr(>F)
Perlakuan      8  586.0    73.25   8.297 0.00019 ***
Kelompok      2   39.2    19.61   2.221 0.14090
Residuals     16  141.3     8.83
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Atau, di python:

```
import pandas as pd

RAKLpy=pd.read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vTi9y0_OJZ5pHMmPRlPqPx
RAKLpy
```

```

    Perlakuan  Kelompok  Produksi
0    Kontrol        1    10.19
1    Kontrol        2     9.26
```

2	Kontrol	3	12.73
3	K2P1	1	32.02
4	K2P1	2	25.76
5	K2P1	3	19.72
6	K2P2	1	23.91
7	K2P2	2	21.99
8	K2P2	3	21.42
9	K2P3	1	17.15
10	K2P3	2	15.66
11	K2P3	3	10.37
12	K2P4	1	10.35
13	K2P4	2	13.31
14	K2P4	3	14.31
15	K3P1	1	21.98
16	K3P1	2	19.43
17	K3P1	3	16.16
18	K3P2	1	18.08
19	K3P2	2	13.50
20	K3P2	3	18.32
21	K3P3	1	18.07
22	K3P3	2	14.01
23	K3P3	3	14.39
24	K3P4	1	12.37
25	K3P4	2	16.32
26	K3P4	3	10.20

Data sudah dalam format yang tepat untuk fungsi `ols`:

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

RAKLlm= ols('Produksi ~ C(Perlakuan)+ C(Kelompok)',data=RAKLpy)
fitRAKL=RAKLlm.fit()
```

Lalu, lakukan ANOVA dari objek fit tersebut:

```
table = sm.stats.anova_lm(fitRAKL)
print(table)
```

	df	sum_sq	mean_sq	F	PR(>F)
C(Perlakuan)	8.0	586.039763	73.254970	8.297279	0.000190
C(Kelompok)	2.0	39.210696	19.605348	2.220615	0.140898

Residual	16.0	141.260704	8.828794	NaN	NaN
----------	------	------------	----------	-----	-----

Hasil ANOVA tersebut relatif sama. Peluang menemukan nilai F yang sama atau lebih besar jika tak ada efek perlakuan sangat kecil - 0.0002 (0.02 persen). Namun, peluang menemukan nilai F yang sama atau lebih besar jika tak ada efek kelompok kira-kira 0.15 (15 persen). Kita mungkin lebih nyaman menyimpulkan bahwa perlakuan berpengaruh daripada bahwa kelompok berpengaruh. Misal, jika kita memakai $\alpha = 5\%$ di kedua kasus, kita akan menolak H_0 untuk perlakuan tapi taktolak H_0 untuk kelompok.

2.3.5 Topik-topik tambahan

2.3.5.1 Efisiensi relatif

Efisiensi relatif RAKL dengan RAL tanpa penyesuaian dihitung sebagai berikut:

$$ER_0 = \frac{\hat{\sigma}_r^2}{\sigma_b^2}$$

Lalu, $\hat{\sigma}_b^2$ adalah penduga ragam galat RAK, yang merupakan KTG.

Penduga ragam galat RAL sedikit lebih kompleks:

$$\sigma_r^2 = \frac{(r-1)KTB + r(t-1)KTG}{tr-1}$$

Penduga ini diturunkan dengan mencari nilai harapan kuadrat tengah dari semua kombinasi rancangan. Pada dasarnya, nilai harapan penduga tersebut akan menjadi KTG RAL. Namun, perbandingan ragam galat tersebut tidak dapat dilakukan secara naif. Sebaran Student t (yang akan digunakan di perbandingan berganda) tidak stabil saat derajat bebas kecil, khususnya di bawah 20. Nilai t kritis akan menurun relatif besar dalam situasi tersebut sehingga probabilitas menemukan perbedaan yang signifikan antara rata-rata perlakuan berfluktuasi. Oleh karena itu, rumus efisiensi relatif memiliki penyesuaian sebagai berikut:

$$ER = \frac{(db_{g[RAK]} + 1)(db_{g[RAL]} + 3)}{(db_{g[RAK]} + 3)(db_{g[RAL]} + 1)} \cdot \frac{\hat{\sigma}_r^2}{\hat{\sigma}_b^2}$$

Di mana $db_{g(RAK)}$ dan $db_{g(RAL)}$ adalah db galat RAK dan RAL. Derajat bebas tersebut diperoleh dengan memakai rumus db galat RAL pada hasil RAK yang kita punya. Dalam contoh sebelumnya, diketahui bahwa db galat adalah 16, dengan 9 perlakuan dan 3 kelompok $(t-1)(r-1) = 8 \cdot 2 = 16$. Derajat bebas galat RAL memiliki rumus $t(r-1) = 9 \cdot 2 = 18$. Atau, dapat memakai rumus $db_{g(RAL)} = db_{g(RAK)} + db_b = (t-1)(r-1) + (r-1) = t(r-1)$.

Untuk mengimplementasikan metode tersebut, buat data frame dari hasil aov terlebih dahulu. Summary aov merupakan sebuah *list*, dan elemen pertama dari elemen pertama list tersebut merupakan tabel ANOVA:

```
#ambil elemen 1 [1]; ambil elemen 1 dari elemen 1 [1] [[1]]
TabelANOVA<-as.data.frame(summary(modelRAKL)[1][[1]])
knitr::kable(TabelANOVA)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Perlakuan	8	586.0398	73.254970	8.297279	0.0001896
Kelompok	2	39.2107	19.605348	2.220614	0.1408977
Residuals	16	141.2607	8.828794	NA	NA

Lalu, ambil nilai-nilai derajat bebas dan penduga ragam yang diinginkan:

```
#Ambil derajat bebas
dbg<-TabelANOVA$Df[3]
dbkel<-TabelANOVA$Df[2]
dbp<-TabelANOVA$Df[1]
r<-dbkel+1
dbg_RAL<-dbg+dbkel

#Ambil kuadrat tengah
KTG<-TabelANOVA$`Mean Sq`[3]
KTB<-TabelANOVA$`Mean Sq`[2]

#hitung penduga ragam RAL
sigmaRAL<-(dbkel*KTB+r*dbp*KTG)/(dbg+dbkel+dbp)
```

Setelah semua nilai terhitung, hitung efisiensi relatif:

```
koreksi<-((dbg+1)*(dbg_RAL+3))/((dbg+3)*(dbg_RAL+1))
RE<-koreksi*(sigmaRAL/KTG)
```

Semua input tersebut dapat digabung menjadi suatu fungsi:

```
efisiensiRAKL<-function(aovariance){
  TabelANOVA<-as.data.frame(summary(aovariance)[1][[1]])
  #Ambil derajat bebas
  dbg<-TabelANOVA$Df[3]
  dbkel<-TabelANOVA$Df[2]
```

```

dbp<-TabelANOVA$Df[1]
r<-dbkel+1
dbg_RAL<-dbg+dbkel

#Ambil kuadrat tengah
KTG<-TabelANOVA$`Mean Sq`[3]
KTB<-TabelANOVA$`Mean Sq`[2]

#hitung penduga ragam RAL
sigmaRAL<-(dbkel*KTB+r*dbp*KTG)/(dbg+dbkel+dbp)

#hitungRE
koreksi<-((dbg+1)*(dbg_RAL+3))/((dbg+3)*(dbg_RAL+1))
RE<-koreksi*(sigmaRAL/KTG)
return(RE)
}
efisiensiRAKL(modelRAKL)

```

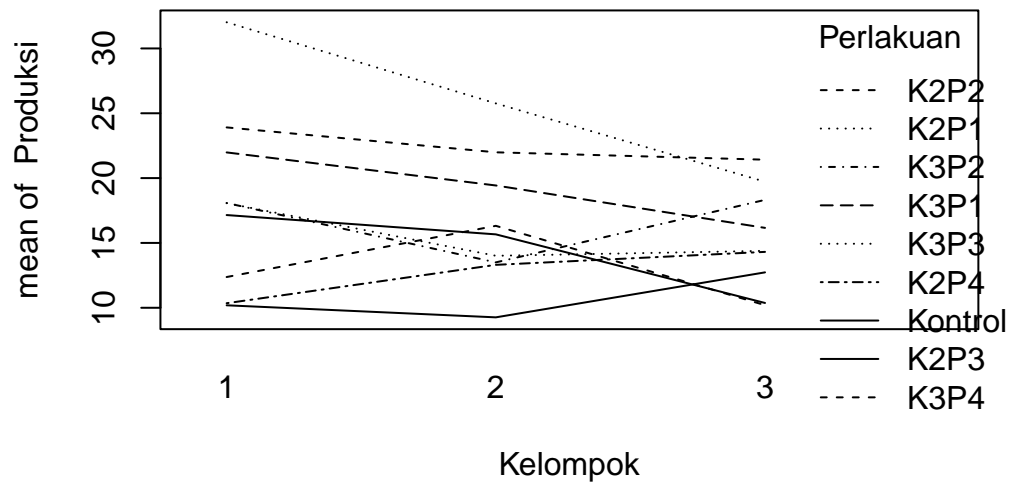
```
[1] 1.081773
```

Interpretasi dari efisiensi relatif tersebut adalah perlu ulangan sebesar RE kali di RAL untuk mencapai performa sama di uji. Dalam kasus ini, perlu 1.08 kali ulangan agar RAL memiliki performa sama dengan RAKL.

2.3.5.2 Plot interaksi

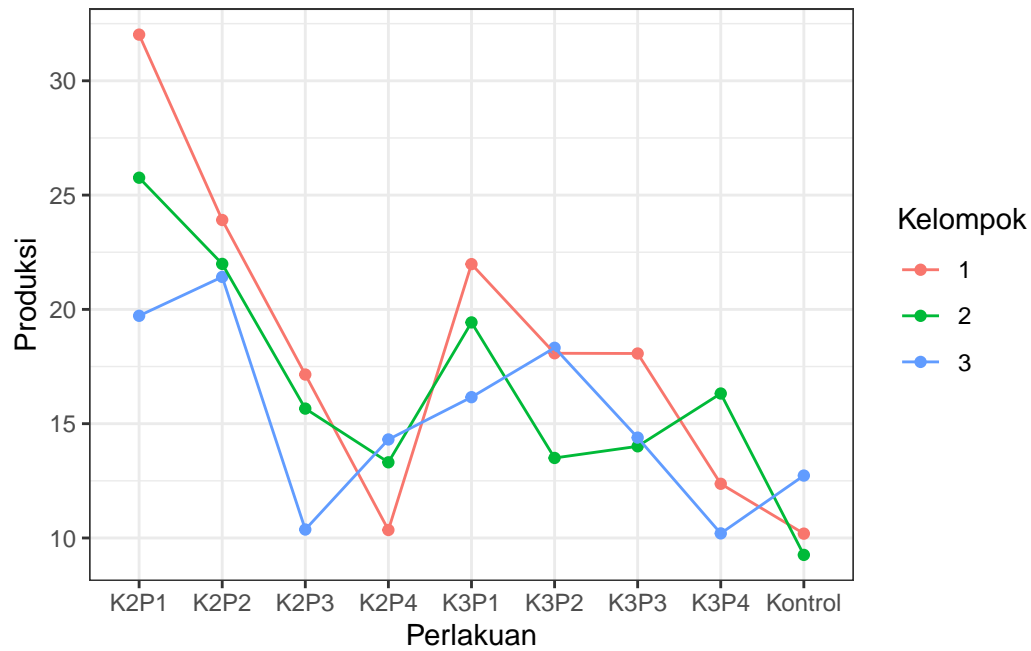
Salah satu asumsi RAKL dengan model tetap adalah tidak adanya interaksi antara perlakuan dan kelompok. Plot interaksi dapat dibuat sebagai berikut:

```
with(DataRAKL,interaction.plot(Kelompok, Perlakuan, Produksi))
```



Atau, jika ingin menggunakan ggplot agar plot lebih rapi:

```
library(ggplot2)
ggplot(DataRAKL, aes(x = Perlakuan, y = Produksi, colour = Kelompok)) +
  geom_point(data = DataRAKL, aes(y = Produksi)) +
  geom_line(data = DataRAKL, aes(y = Produksi, group = Kelompok)) +
  theme_bw()
```

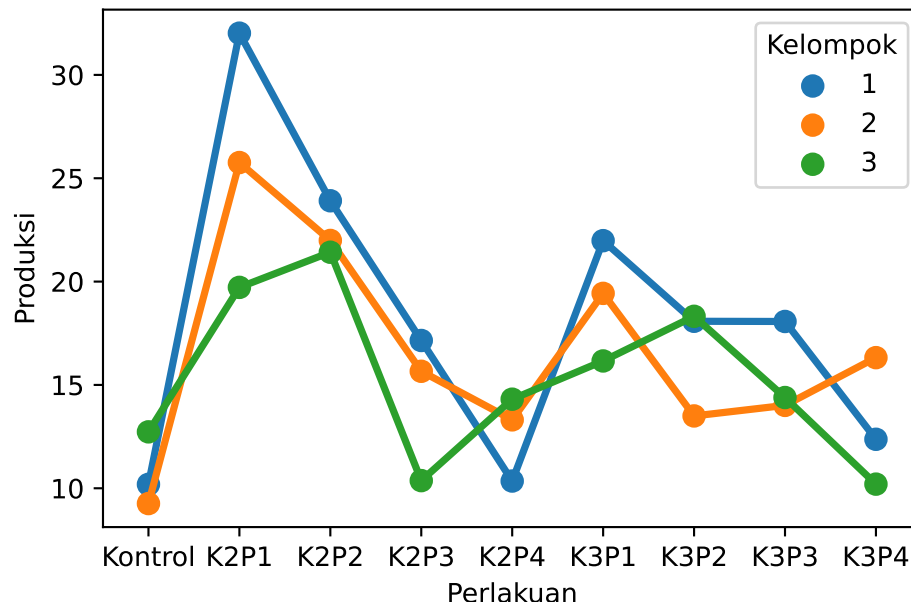


Plot interaksi juga dapat dibuat di python. Pastikan data frame RAKLpy sudah di-load terlebih dahulu sebelum melakukan plotting:

```
import seaborn as sns
import matplotlib.pyplot as plt

g=sns.pointplot(x = "Perlakuan",
                y = "Produksi",
                hue= "Kelompok",
                data = RAKLpy)

# show the plot
plt.show()
```

Terlihat beberapa perbedaan antara pola per kelompok, khususnya setelah K3P1 di kedua plot. Pola-pola tersebut menandakan ada kemungkinan interaksi yang perlu diuji lebih lanjut.

2.3.5.3 Pendugaan Data Hilang

Data hilang diduga dengan MKT. Nilai data hilang dicari sehingga meminumkan JKG. Oleh karena itu, rumus adalah:

$$y_{ij} = \frac{ty'_{i.} + ry'_{.j} - y'_{..}}{(t-1)(r-1)}$$

Dengan t adalah jumlah perlakuan, $y'_{i.}$ total data perlakuan data hilang tersebut, r jumlah kelompok, $y'_{.j}$ total data suatu kelompok. Package R yang memiliki fitur estimasi data hilang adalah `st4gi`. Package tersebut tidak bisa didapat melalui `install.packages` sehingga cara instalasi package sebagai berikut:

```
devtools::install_github("reyzaguirre/st4gi")
```

Devtools digunakan untuk menginstalasi dari GitHub. Seperti `agricolae`, package `st4gi` juga menyediakan pembuatan bagan percobaan, dengan command `cr.rcbd`. Pembuatan bagan percobaan dengan `st4gi` adalah sebagai berikut:

```
library(st4gi)
dfrakl<-cr.rcbd(geno=levels(DataRAKL$Perlakuan), #perlakuan
               nb=3 #jumlah kelompok
               )
dfrakl<-dfrakl$book #data frame dari bagan percobaan
knitr::kable(head(dfrakl,n=10))
```

plot.num	block	row	col	geno
1	1	1	1	Kontrol
2	1	1	2	K2P2
3	1	1	3	K3P4
4	1	2	3	K3P3
5	1	2	2	K2P3
6	1	2	1	K2P1
7	1	3	1	K2P4
8	1	3	2	K3P1
9	1	3	3	K3P2
10	2	1	1	K2P2

Argumen geno (genotipe) digunakan karena package st4gi digunakan dalam implementasi riset genetik. Data frame yang dibuat dari bagan percobaan akan digunakan sebagai input estimasi data hilang. Sebelumnya, telah dicoba input estimasi data hilang menggunakan DataRAKL asli yang tidak dibuat dari bagan percobaan st4gi, tetapi gagal. Oleh karena itu, input data dari DataRAKL ke dfr (hasil st4gi). Harus dipastikan bahwa data tersebut memiliki urutan sama dengan sorting. Sorting tersebut dimulai dari kelompok, lalu perlakuan:

```
dforder<-dfrakl[order(dfrakl$block, dfrakl$geno),] #order data frame hasil st4gi
RAKLorder<-DataRAKL[order(DataRAKL$Kelompok, DataRAKL$Perlakuan),] #order data frame RAKL
knitr::kable(head(dforder,n=12))
```

plot.num	block	row	col	geno	
6	6	1	2	1	K2P1
2	2	1	1	2	K2P2
5	5	1	2	2	K2P3
7	7	1	3	1	K2P4
8	8	1	3	2	K3P1
9	9	1	3	3	K3P2
4	4	1	2	3	K3P3
3	3	1	1	3	K3P4

	plot.num	block	row	col	geno
1	1	1	1	1	Kontrol
15	15	2	2	1	K2P1
10	10	2	1	1	K2P2
13	13	2	2	3	K2P3

Bandingkan dengan RAKL order:

```
knitr::kable(head(RAKLorder,n=12))
```

Perlakuan	Kelompok	Produksi
K2P1	1	32.02
K2P2	1	23.91
K2P3	1	17.15
K2P4	1	10.35
K3P1	1	21.98
K3P2	1	18.08
K3P3	1	18.07
K3P4	1	12.37
Kontrol	1	10.19
K2P1	2	25.76
K2P2	2	21.99
K2P3	2	15.66

Kedua data frame tersebut memiliki urutan mirip. Masukkan data dari hasil percobaan ke bagan percobaan st4gi dan hilangkan suatu data:

```
dforder$y<-RAKLorder$Produksi #input data respon hasil sort
dforder[12,'y']<-NA #buat salah satu data NA
```

Lalu, cek data hilang tersebut. Urutan argumen fungsi adalah peubah respon, perlakuan, kelompok, dan data frame:

```
cek<-ck.rcbd('y', 'geno', 'block', dforder) #peubah respon, perlakuan, kelompok, df
knitr::kable(cek$tfr)
```

	1	2	3
K2P1	1	1	1

	1	2	3
K2P2	1	1	1
K2P3	1	0	1
K2P4	1	1	1
K3P1	1	1	1
K3P2	1	1	1
K3P3	1	1	1
K3P4	1	1	1
Kontrol	1	1	1

Data hilang berada di perlakuan K2P3 dan kelompok 2. Kita akan bandingkan hasil perhitungan manual dengan fungsi R. Pertama, cari $y'_{i.}$ dan $y'_{.j}$. Buat tabel total perlakuan dengan dplyr. Operator `%>%` merupakan pipa yang melanjutkan output suatu fungsi ke fungsi sebelumnya. Misal `dforder %>%` berarti data.frame `dforder` menjadi input fungsi selanjutnya, yaitu pengelompokan `group_by(geno)`. Lalu hasilnya dibuatkan tabel simpulan, yaitu dengan `sum`:

```
library(dplyr)
```

Attaching package: 'dplyr'

The following object is masked from 'package:edibble':

```
as_data_frame
```

The following objects are masked from 'package:stats':

```
filter, lag
```

The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

```
TotalPerlakuan<-dforder %>%
  group_by(geno) %>%
  summarise(Freq = sum(y,na.rm=T))
knitr::kable(TotalPerlakuan)
```

geno	Freq
K2P1	77.50
K2P2	67.32
K2P3	27.52
K2P4	37.97
K3P1	57.57
K3P2	49.90
K3P3	46.47
K3P4	38.89
Kontrol	32.18

Dan buat tabel total kelompok:

```
TotalKelompok<-dforder %>%
  group_by(block) %>%
  summarise(Freq = sum(y,na.rm=T))
knitr::kable(TotalKelompok)
```

block	Freq
1	164.12
2	133.58
3	137.62

Ambil total perlakuan K2P3, total kelompok 2, dan total nilai peubah respon:

```
Total<-sum(dforder$y,na.rm=T)
TotalK2P3<-TotalPerlakuan$Freq[3]
Total2<-TotalKelompok$Freq[2]
(9*TotalK2P3+3*Total2-Total)/16
```

```
[1] 13.31875
```

Bandingkan dengan estimasi menggunakan fungsi R. Urutan argumen sama dengan argumen saat pengecekan nilai hilang di RAKL (respon, perlakuan, kelompok, data frame):

```
knitr::kable(mve.rcbd('y','geno','block',dforder)[12,])
```

	geno	block	y	y.est
13	K2P3	2	NA	13.31875

Dapat dilihat bahwa hasil pengisian nilai hilang sama dengan hasil penghitungan manual.

2.3.5.4 Koefisien Keragaman

Koefisien Keragaman merupakan ukuran keheterogenan unit percobaan. Rumus koefisien tersebut cukup sederhana, yaitu penduga simpangan baku dibagi penduga nilai harapan. Karena penduga ragam adalah KTG, penduga simpangan baku adalah akar dari KTG. Rumus tersebut menjadi:

$$KK = \frac{\hat{\sigma}}{\bar{y}_{..}} \cdot 100\% = \frac{\sqrt{KTG}}{\bar{y}_{..}} \cdot 100\%$$

Untuk menghitung KK dari data RAKL relatif sederhana. Note bahwa dalam prosedur estimasi nilai hilang sudah ditemukan KTG, jadi KTG hanya perlu dipanggil. Jika belum ditemukan, ikuti prosedur yang ada di bagian estimasi nilai hilang:

```
sqrt(KTG)/mean(DataRAKL$Produksi)*100
```

```
[1] 17.78923
```

Nilai tersebut memiliki interpretasi besar simpangan baku contoh adalah 17 persen dari besar rata-rata contoh.

2.3.5.5 Jumlah ulangan

Jumlah kelompok dalam RAKL dihitung menggunakan beberapa kaidah:

1. Minimal 2
2. Minimal derajat bebas galat 12
3. Formula perbedaan rata-rata maksimal

Kaidah menggunakan db galat dapat diturunkan sebagai berikut:

$$\begin{aligned}db_g &= (t-1)(r-1) \geq 12 \\ r-1 &\geq \frac{12}{t-1} \\ r &\geq \frac{12}{t-1} + 1\end{aligned}$$

Dan implementasi rumus tersebut di R diperlihatkan di bawah:

```
t<-9 #nilai t dapat disesuaikan
ceiling((12/(t-1))+1)
```

```
[1] 3
```

Seperti di bab pertama, ceiling digunakan karena pembulatan biasa dapat menghasilkan nilai yang lebih kecil dari hasil rumus. Pembulatan ke atas mencegah hal tersebut. Maka, dibutuhkan minimal 3 kelompok dalam percobaan ini.

Formula perhitungan jumlah kelompok adalah

$$r \geq \frac{2 \cdot t_{\alpha/2;dbg}^2 \cdot \sigma^2}{\delta^2}$$

Asal dari formula tersebut adalah BNT. Formula tersebut menyatakan jumlah kelompok yang dibutuhkan untuk mendeteksi perbedaan sebesar δ antara rata-rata dua perlakuan di tingkat signifikansi $\alpha\%$. σ^2 menandakan ragam, dan $t_{\alpha/2;dbg}^2$ berkaitan dengan sebaran t yang digunakan di BNT. Alternatif formula tersebut adalah:

$$r \geq \frac{2 \cdot t_{\alpha/2;dbg}^2 \cdot kk^2}{q^2}$$

Di mana KK adalah koefisien keragaman dan q adalah ukuran perbedaan yang dinyatakan dalam persentase. Perhitungan formula tersebut relatif mudah diimplementasikan di R. Oleh karena itu, hanya akan ditunjukkan cara mendapatkan tabel-t dari R. Lower.tail dibuat FALSE agar nilai tabel t tidak minus.

```
qt(p=0.025,df=16,lower.tail=F) #alpha atau alpha/2, dbg
```

```
[1] 2.119905
```

2.4 RBSL

RBSL merupakan rancangan yang memungkinkan adanya dua faktor pengganggu. Di luar faktor pengganggu tersebut dan perlakuan, unit percobaan dianggap homogen. RBSL memiliki batasan yaitu jumlah perlakuan = baris = lajur. Ini berarti, taraf-taraf dari faktor pengganggu juga harus sama dengan taraf faktor perlakuan. Oleh karena itu, aplikasi RBSL relatif terbatas saat jumlah faktor-faktor tersebut dapat bervariasi.

Lalu, satu perlakuan hanya muncul sekali di satu baris dan lajur. Oleh karena hal tersebut dan batasan jumlah perlakuan, baris, dan lajur pada RBSL, terdapat r^2 unit percobaan, dengan r adalah jumlah perlakuan, baris, atau lajur. Ini berbeda dengan perlakuan diulang di semua kombinasi baris dan lajur. Jika perlakuan diulang di semua kombinasi baris dan lajur, perlu $r^2 \cdot t$ unit percobaan.

Jumlah unit percobaan tersebut akan menjadi sangat besar jika perlakuan hendak ditambah. Dengan r kelompok atau ulangan, hanya perlu r unit percobaan tambahan untuk menambah satu perlakuan. Sedangkan, di RBSL, $(r+1)^2 - r^2 = r^2 + 2r + 1 - r^2 = 2r + 1$, perlu $2r + 1$ unit percobaan tambahan. Jumlah tersebut akan menjadi sangat besar jika perlakuan makin banyak.

Selain itu, rumus derajat bebas galat bagi RBSL adalah $(r-1)(r-2)$. Ini memiliki beberapa implikasi. Pertama, jumlah minimal perlakuan adalah dua karena jika tidak derajat bebas galat akan nol. KTG akan menjadi tak terdefinisi. Dalam kasus perlakuan lebih dari dua, tetapi masih sedikit, db galat akan kecil.

Selain itu, analisis akan kompleks jika ada data hilang atau salah penempatan baris. Diperlukan juga asumsi **tidak ada interaksi**. Pengaruh baris, perlakuan, dan lajur sama saja bagaimanapun kombinasinya.

Akan dianalisis kasus RBSL sebagai berikut:

Percobaan tentang pengaruh pemberian pupuk dengan dosis yang berbeda (100, 200, 300, 400) dilakukan di daerah lereng pegunungan. Sumber keragaman unit percobaan secara garis besar dapat diklasifikasikan menjadi dua, yaitu kemiringan lahan dan arah irigasi. Kemiringan lahan dan arah irigasi dapat dikelompokkan masing-masing menjadi empat kelompok.

2.4.1 Pengacakan

Algoritme pengacakan RBSL sebagai berikut:

1. Susun perlakuan pada arah diagonal secara acak.
2. Acak penempatan baris.
3. Acak penempatan lajur.

Algoritme tersebut dapat diterapkan di R menggunakan package agricolae atau package st4gi. Penerapan agricolae sebaga berikut:

```
library(agricolae)
perlakuanBSL<-seq(100,400,100)
baganBSL<-design.lsd(trt=perlakuanBSL) #masukkan perlakuan
knitr::kable(baganBSL$sketch)
```

400	200	300	100
200	400	100	300
300	100	200	400
100	300	400	200

Sedangkan, penerapan st4gi sebagai berikut:

```
library(st4gi)
baganBSL2<-cr.lsd(geno=perlakuanBSL) #geno=perlakuan
knitr::kable(baganBSL2$book)
```

plot.num	row	col	geno
1	1	1	300
2	1	2	400
3	1	3	100
4	1	4	200
5	2	4	300
6	2	3	200
7	2	2	100
8	2	1	400
9	3	1	200
10	3	2	300
11	3	3	400
12	3	4	100
13	4	4	400
14	4	3	300
15	4	2	200
16	4	1	100

Jika dilihat, perlakuan hanya muncul sekali di tiap baris dan di tiap lajur, sesuai batasan pengacakan yang dijelaskan sebelumnya.

2.4.1.1 Pengacakan edible

Dalam RBSL, ada lajur dan baris. Dalam kasus ini lajur dan baris adalah kemiringan dan lahan. Lalu, tiap petak berada di antara suatu kombinasi kemiringan dan lahan. Oleh karena itu, gunakan `crossed_by(kemiringan,lahan)` dalam pendefinisian petak.

```
library(edibble)

desRBSL<-design(name="Pengaruh dosis pupuk di lereng") %>%
  set_units(kemiringan=4,
            irigasi=4,
            petak=crossed_by(kemiringan,irigasi)) %>%
  set_trts(dosis=perlakuanBSL) %>%
  allot_trts(dosis ~ petak) %>%
  assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desRBSL,n=10))
```

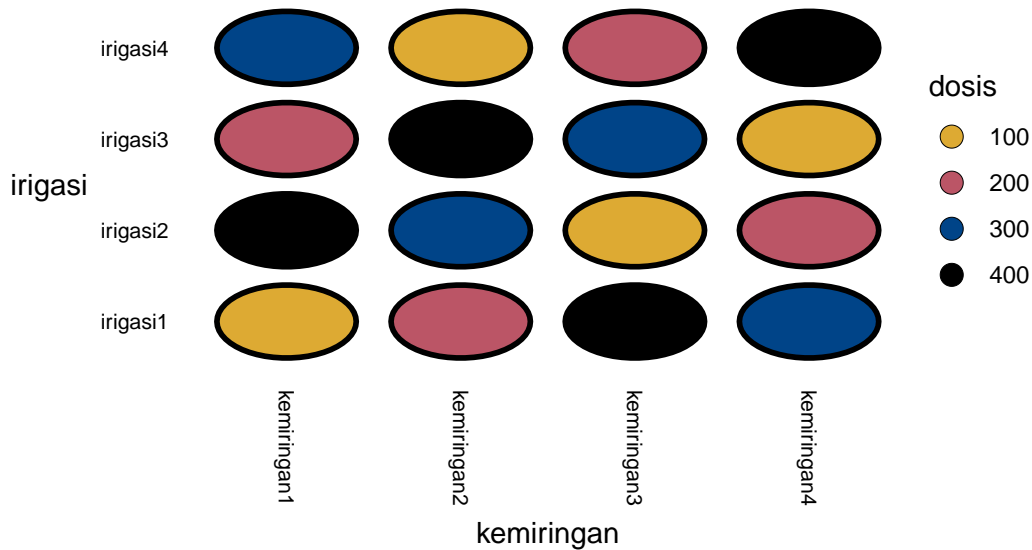
kemiringan	irigasi	petak	dosis
kemiringan1	irigasi1	petak1	100
kemiringan2	irigasi1	petak2	200
kemiringan3	irigasi1	petak3	400
kemiringan4	irigasi1	petak4	300
kemiringan1	irigasi2	petak5	400
kemiringan2	irigasi2	petak6	300
kemiringan3	irigasi2	petak7	100
kemiringan4	irigasi2	petak8	200
kemiringan1	irigasi3	petak9	200
kemiringan2	irigasi3	petak10	400

Hasil pengacakan dapat digambarkan:

```
degust::autoplot(desRBSL)
```

Pengaruh dosis pupuk di lereng

Unit: petak



Bandingkan dengan menu `lsd`:

```
lsd <- takeout(menu_lsd(t = 4, seed=420))
examine_recipe(lsd)
```

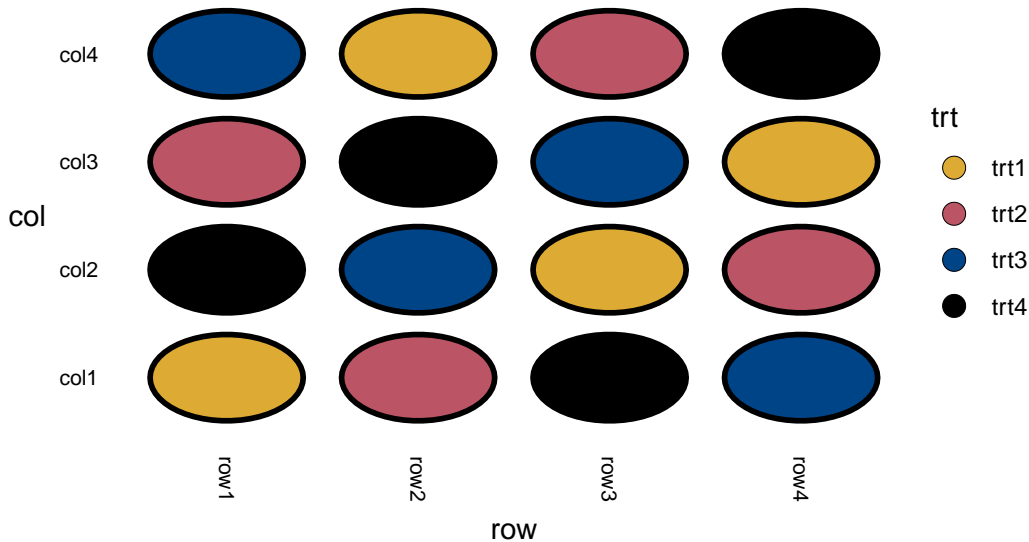
```
design("Latin Square Design") %>%
  set_units(row = 4,
            col = 4,
            unit = crossed_by(row, col)) %>%
  set_trts(trt = 4) %>%
  allot_trts(trt ~ unit) %>%
  assign_trts("random", seed = 420) %>%
  serve_table()
```

Pembuatan rancangan sama. Plot juga sama:

```
degust::autoplot(lsd)
```

Latin Square Design

Unit: unit



2.4.2 Model

Model linier aditif dari RBSL adalah:

$$Y_{ij(k)} = \mu + \alpha_i + \beta_j + \tau_k + \varepsilon_{ij(k)}$$

dengan keterangan: $i = 1, 2, \dots, r$; $j = 1, 2, \dots, r$; $k = 1, 2, \dots, r$ (dalam kata lain, jumlah perlakuan, lajur, dan baris sama) ; $\varepsilon_{ij(k)} \sim N(0, \sigma^2)$, dan:

1. $Y_{ij(k)}$ = nilai pengamatan pada perlakuan ke-k dalam baris ke-i, lajur ke-j
2. μ = rata-rata umum
3. α_i = pengaruh baris ke-i
4. β_j = pengaruh lajur ke-j
5. τ_k = pengaruh perlakuan ke-k dalam baris ke-i, lajur ke-j
6. $\varepsilon_{ij(k)}$ = pengaruh acak pada perlakuan ke-k dalam baris ke-i, lajur ke-j

Asumsi sama seperti RAKL, ditambah $\sum_{i=1}^t \alpha_i = 0$ untuk model tetap dan $\alpha_i \sim N(0, \sigma_\alpha^2)$ untuk model acak. α_i diduga $\bar{y}_{i.} - \bar{y}_{..}$ dari hasil penurunan MKT.

2.4.3 ANOVA

Hipotesis yang diuji sama dengan RAKL, dengan tambahan hipotesis mengenai pengaruh baris. Hipotesis pertama adalah mengenai **pengaruh perlakuan**:

$$H_0 : \tau_1 = \dots = \tau_r = 0$$

$$H_1 : \text{Setidaknya ada satu } i \text{ di mana } \tau_i \neq 0$$

Yaitu, bahwa semua perlakuan tidak berpengaruh atau sebaliknya, terdapat sedikitnya satu perlakuan yang berpengaruh. Lalu, itu, ada hipotesis mengenai **pengaruh lajur**, atau kelompok:

$$H_0 : \beta_1 = \dots = \beta_r = 0$$

$$H_1 : \text{Setidaknya ada satu } j \text{ di mana } \beta_j \neq 0$$

Dengan interpretasi yang sama seperti hipotesis sebelumnya. Hipotesis yang ditambah RBSL adalah pengaruh baris:

$$H_0 : \alpha_1 = \dots = \alpha_r = 0$$

$$H_1 : \text{Setidaknya ada satu } i \text{ di mana } \alpha_i \neq 0$$

Hipotesis tersebut diuji dengan uji-F, yang disimpulkan di tabel sidik ragam:

```
cat('
| Sumber Keragaman| db| JK| KT| F-hit|F(dbP,dbG)|
|-----:|-----:|-----:|-----:|-----:|
|      Perlakuan|      r-1|      JKP| JKP/dbP | KTP/KTG | |
|      Baris|      r-1|      JKB| JKB/dbB | KTB/KTG | |
|      Lajur|      r-1|      JKL| JKL/dbL | KTL/KTG | |
|      Galat|      (r-1)(r-2)|      JKG| JKG/dbG | | |
|      Total|      r*r-1|      JKT| | | |')
```

```
| Sumber Keragaman| db| JK| KT| F-hit|F(dbP,dbG)|
|-----:|-----:|-----:|-----:|-----:|
|      Perlakuan|      r-1|      JKP| JKP/dbP | KTP/KTG | |
|      Baris|      r-1|      JKB| JKB/dbB | KTB/KTG | |
|      Lajur|      r-1|      JKL| JKL/dbL | KTL/KTG | |
|      Galat|      (r-1)(r-2)|      JKG| JKG/dbG | | |
|      Total|      r*r-1|      JKT| | | |
```

Rumus-rumus jumlah kuadrat di tabel sidik ragam tersebut adalah:

$$\begin{aligned}
FK &= \frac{y_{..}^2}{r^2} \\
JK_T &= \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^r [y_{ij(k)} - \bar{y}_{..(.)}]^2 = \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^r y_{ij(k)}^2 - FK \\
JK_P &= \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^r [\bar{y}_{..(k)} - \bar{y}_{..(.)}]^2 = \sum_{k=1}^r \frac{y_{..(k)}^2}{r} - FK \\
JK_B &= \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^r [\bar{y}_{i.(.)} - \bar{y}_{..(.)}]^2 = \sum_{i=1}^r \frac{y_{i.(.)}^2}{r} - FK \\
JK_L &= \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^r [\bar{y}_{.j(.)} - \bar{y}_{..(.)}]^2 = \sum_{j=1}^r \frac{y_{.j(.)}^2}{r} - FK \\
JK_G &= JK_T - JK_P - JK_B - JK_L
\end{aligned}$$

Pada intinya, jumlah kuadrat tiap faktor (perlakuan, baris, atau lajur) dihitung dengan mencari total semua observasi dengan taraf-taraf tertentu dari faktor tersebut. Lalu, total tersebut dikuadratkan dan dibagi jumlah perlakuan, baris, atau lajur, dan hasil tersebut dijumlahkan sesuai jumlah tersebut.

2.4.4 Implementasi R dan Python

Relatif jelas bahwa rancangan yang tepat digunakan adalah RBSL. [Data](#) ditautkan. Analisis dimulai dengan membaca data tersebut:

```
library(google Sheets4)
dataRBSL<-read_sheet("https://docs.google.com/spreadsheets/d/1c9dvm4p747hX9nS3y6mMSCgGSoJM")
```

```
v Reading from "RBSL_P2 Rancob".
```

```
v Range 'Sheet1'.
```

```
knitr::kable(dataRBSL)
```

Miring	Irigasi	Pupuk	Produksi
M1	I1	P1	5.6
M1	I2	P3	4.3
M1	I3	P2	4.8

Miring	Irigasi	Pupuk	Produksi
M1	I4	P4	5.3
M2	I1	P2	5.9
M2	I2	P1	6.0
M2	I3	P4	5.8
M2	I4	P3	5.2
M3	I1	P3	4.3
M3	I2	P4	4.9
M3	I3	P1	4.8
M3	I4	P2	5.2
M4	I1	P4	3.9
M4	I2	P2	4.9
M4	I3	P3	3.8
M4	I4	P1	4.7

Setelah memastikan perlakuan, lajur, dan baris berbentuk faktor, tinggal dilakukan anova. Penambahan faktor baru di ANOVA relatif mudah, yaitu dengan +.

```
dataRBSL$Miring<-as.factor(dataRBSL$Miring)
dataRBSL$Irigasi<-as.factor(dataRBSL$Irigasi)
dataRBSL$Pupuk<-as.factor(dataRBSL$Pupuk)
RBSLAoV<-aov(Produksi~Miring+Irigasi+Pupuk,dataRBSL) #respon~perlakuan+baris+lajur,dataset
summary(RBSLAoV)
```

```

              Df Sum Sq Mean Sq F value    Pr(>F)
Miring         3  4.062   1.3542   12.897 0.00502 **
Irigasi        3  0.203   0.0675    0.643 0.61504
Pupuk          3  1.882   0.6275    5.976 0.03106 *
Residuals     6  0.630   0.1050
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gunakan statsmodels untuk analisis di python:

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols

RBSLpy=pd.read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vQySAMP0lmbObWHA9ZQS80
RBSLlm= ols('Produksi ~ C(Irigasi)+C(Miring)+C(Pupuk)',data=RBSLpy).fit() #C=peubah kategori
```

```
table = sm.stats.anova_lm(RBSLlm, typ=2) # Type 2 ANOVA DataFrame
print(table)
```

	sum_sq	df	F	PR(>F)
C(Irigasi)	0.2025	3.0	0.642857	0.615037
C(Miring)	4.0625	3.0	12.896825	0.005020
C(Pupuk)	1.8825	3.0	5.976190	0.031062
Residual	0.6300	6.0	NaN	NaN

Analisis diawali dengan membaca CSV melalui pandas. C(faktor) digunakan agar faktor dalam percobaan dianggap peubah kategorik. Hasil dari ANOVA tersebut menunjukkan bahwa Pupuk memiliki p-value sekitar 3%, yang berarti kemungkinan menemukan nilai-F tersebut jika H_0 benar sebesar 3%. Jika digunakan $\alpha = 5\%$, H_0 akan ditolak dan setidaknya salah satu pengaruh perlakuan (Pupuk) tidak nol. P-value kemiringan lebih kecil lagi. Dengan prosedur sama, dapat diinterpretasikan bahwa setidaknya salah satu pengaruh kemiringan tidak nol. Namun, belum cukup bukti untuk menyatakan bahwa setidaknya salah satu pengaruh irigasi tidak nol.

2.4.5 Topik Tambahan

2.4.5.1 Pendugaan Nilai Hilang

Rumus pendugaan nilai hilang di RBSL adalah sebagai berikut:

$$\hat{y}_{ij(k)} = \frac{p(y_{i.} + y_{.j} + T_{(k)}) - 2y_{..}}{(p-1)(p-2)}$$

Nilai-nilai yang diperlukan dapat diekstrak manual dan dihitung, sama seperti sebelumnya.

2.4.5.2 Plot interaksi

Plot interaksi dapat dibuat dengan cara sama seperti RAKL.

3 Perbandingan Nilai Tengah Perlakuan

ANOVA pada RAK, RAKL, dan RBSL hanya dapat menguji ketidaknolannya efek perlakuan. Semisal diketahui setidaknya salah satu perlakuan tersebut memiliki efek, lalu bagaimana? Tentu ada pertanyaan-pertanyaan lain yang menarik, misal:

1. Perlakuan mana yang memberi efek terbesar atau terkecil?
2. Perlakuan mana yang memberi efek mirip?
3. Apakah perlakuan dapat dikelompokkan?

Dan pertanyaan-pertanyaan lain. Pertanyaan itu dijawab **uji lanjut**.

3.1 LSD/BNT

Uji BNT menguji perlakuan secara berpasangan. Menurut Carmer dan Swanson (1973) di Montgomery (2017), BNT efektif mendeteksi perbedaan rata-rata perlakuan jika diaplikasikan ****setelah**** uji-F di ANOVA signifikan di taraf 5 persen. Hipotesis perbandingan BNT sebagai berikut:

$$H_0 : \mu_i = \mu_j$$

$$H_1 : \mu_i \neq \mu_j$$

Dengan nilai kritis BNT sebagai berikut:

$$LSD = t_{\alpha/2, db_g} S_{\bar{y}_{i.} - \bar{y}_{j.}}$$
$$S_{\bar{y}_{i.} - \bar{y}_{j.}} = \sqrt{KTG \left(\frac{1}{r_i} + \frac{1}{r_j} \right)}$$

Jika ulangan sama, $S_{\bar{y}_{i.} - \bar{y}_{j.}} = \sqrt{\frac{2KTG}{r}}$ (kenapa?). Prosedur uji tersebut adalah membandingkan $\bar{y}_{i.} - \bar{y}_{j.}$ dengan nilai kritis. Implementasi di R (dari data RAL sebelumnya) sebagai berikut:

```
library(googleSheets4)
#baca sheet
```

```
DataRAL<-read_sheet("https://docs.google.com/spreadsheets/d/1Bzm_R2Zd4Zbij7B07LGDJW83DU6T3
```

! Using an auto-discovered, cached token.

To suppress this message, modify your code or options to clearly consent to the use of a cached token.

See gargle's "Non-interactive auth" vignette for more details:

<<https://gargle.r-lib.org/articles/non-interactive-auth.html>>

i The googlesheets4 package is using a cached token for 'muh.ammarsahab@apps.ipb.ac.id'.

v Reading from "DataRAL".

v Range 'Sheet1'.

```
#reshaping
library(reshape2)

RALMelt<- melt(DataRAL,
               #variabel yang membedakan tiap baris di tabulasi asli:
               id.vars=c("Dosis"),

               #kolom yang ingin digabung jadi 1
               measured.vars=as.character(seq(1,8)),
               value.name="Perkecambahan")
colnames(RALMelt)[2]<-"Ulangan"
RALMelt$Dosis<-as.factor(RALMelt$Dosis)

#anova
aov_RAL <- aov(Perkecambahan ~ Dosis, data = RALMelt)
summary(aov_RAL)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Dosis	4	2556.4	639.1	51.36	3.6e-14 ***
Residuals	35	435.5	12.4		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

		Rata-Rata per Taraf	Kelompok
	0	100.00	a
LSD	16	100.00	a
3.580545	32	90.25	b
	64	85.00	c
	48	80.00	d

		Rata-Rata per Taraf	Kelompok
	0	100.00	a
LSD	16	100.00	a
3.574374	32	90.25	b
	64	85.00	c
	48	80.00	d

Gunakan library agricolae:

```
library(agricolae)
LSD<-agricolae::LSD.test(aov_RAL, "Dosis") #hasil anova, perlakuan

knitr::kables(
  list(
    knitr::kable(LSD$statistics[6]), #tabel 1
    knitr::kable(LSD$groups, col.names = c("Rata-Rata per Taraf",
                                           "Kelompok")) #tabel 2
  ),
)
```

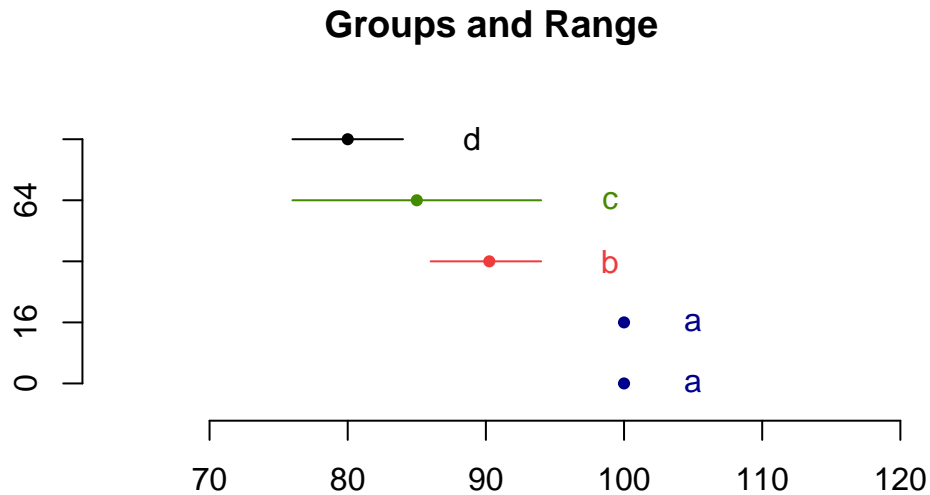
Atau susun argumen sebagai berikut,

```
library(agricolae)
LSD2<-agricolae::LSD.test(RALMelt$Perkecambahan, RALMelt$Dosis,35,12.4) #respon, perlakuan

knitr::kables(
  list(
    knitr::kable(LSD2$statistics[6]) , #tabel 1
    knitr::kable(LSD2$groups, col.names =c("Rata-Rata per Taraf",
                                           "Kelompok")) #tabel 2
  ),
)
```

Dapat dicek sendiri bahwa pengelompokan tersebut benar. Dapat juga dibuat visualisasi:

```
plot(LSD,horiz=T)
```



Note, BNT menguji *semua* perlakuan secara berpasangan, sehingga tidak bisa memilih perbandingan tertentu saja (misal, rata-rata terbesar vs terkecil). Selain itu, BNT memiliki kelemahan, yaitu hanya dapat mengontrol tingkat kesalahan uji sepasang rata-rata saja. Jika ada beberapa pasang rata-rata, tingkat kesalahan berpotensi lebih besar. Banyak perbandingan $g = k(k-1)/2$ (dari permutasi), sehingga *family error rate* sebesar $1-(1-\alpha)^g$. Perhitungan tersebut hanya merupakan kebalikan dari perhitungan $P(\text{Tolak } H_0 | H_0 \text{ benar})$. $(1 - \alpha)$. Jika diasumsikan perbandingan saling bebas, maka probabilitas tersebut menjadi $(1 - \alpha)^g$.

3.2 BNJ

Lalu, bagaimana cari membuat uji yang mengontrol *family error rate*? BNJ mengontrol kesalahan di t buah perlakuan sebesar α , sehingga di tiap pasangan akan menerima kesalahan sebesar $\alpha/2g$ karena uji dua arah menggunakan $\alpha/2$ dan ada g ulangan. Jika jumlah perlakuan banyak, metode ini akan sangat ekstrim. Oleh karena itu, walaupun metode ini tidak terlalu sensitif, metode ini memisahkan perlakuan yang benar-benar beda. BNJ memiliki nilai kritis:

$$BNJ = q_{\alpha;t;db_g} S_{\bar{y}}$$

$$S_{\bar{y}} = \sqrt{KTG/r}$$

Dengan t jumlah perlakuan, q memiliki asal statistik *studentized range*. Jika ulangan tak sama, r didekati:

$$r_h = \frac{t}{\sum_{i=1}^t 1/r_i}$$

Dengan langkah-langkah uji:

1. Urutkan rata-rata perlakuan dari yang terkecil sampai yang terbesar atau sebaliknya
2. Nilai awal $i=1$ (dari kiri) dan $j=1$ (dari kanan)
3. Hitung beda antara rata-rata perlakuan terkecil ke- i dengan terbesar ke- j kemudian bandingkan dengan nilai BNJ, jika beda rata-rata perlakuan lebih kecil lanjutkan ke langkah 5 (tolak H_0) dan jika tidak lanjutkan ke langkah 4
4. Berikan $j = j+1$, jika $j < p$ kembali ke langkah 3
5. Buatlah garis mulai rata-rata perlakuan ke- i sampai ke perlakuan ke- j
6. Berikan $i = i+1$, jika $i < p$ kembali ke langkah 3
7. Stop

Implementasi uji tersebut di R adalah:

```
tukey <- TukeyHSD(aov_RAL, 'Dosis', conf.level = 0.95)

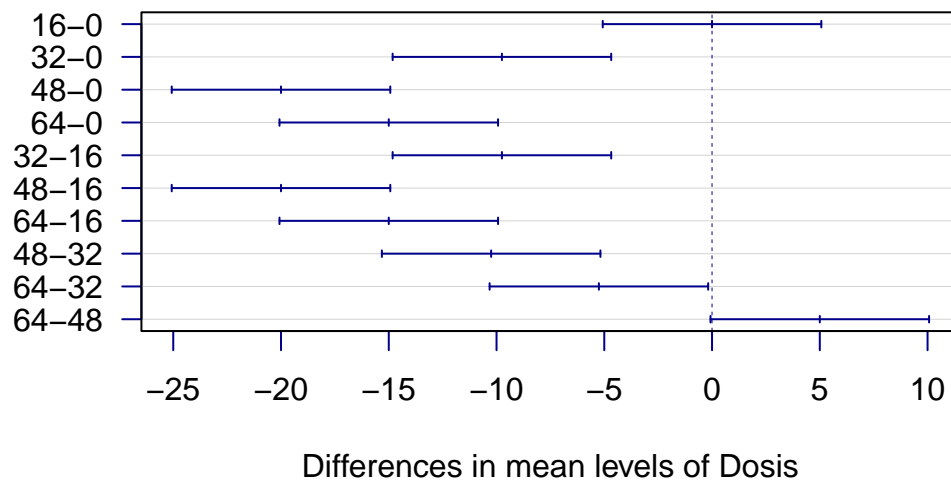
library(kableExtra)
knitr::kable(tukey$Dosis, col.names = c(" ",
                                         " Bawah",
                                         " Atas",
                                         " ")) %>%
add_header_above(c(" Taraf"=1, "Beda"=1, "Batas Selang Kepercayaan"=2, "P-Value"=1))
```

Taraf	Beda	Batas Selang Kepercayaan		P-Value
		Bawah	Atas	
16-0	0.00	-5.0708064	5.0708064	1.0000000
32-0	-9.75	-14.8208064	-4.6791936	0.0000306
48-0	-20.00	-25.0708064	-14.9291936	0.0000000
64-0	-15.00	-20.0708064	-9.9291936	0.0000000
32-16	-9.75	-14.8208064	-4.6791936	0.0000306
48-16	-20.00	-25.0708064	-14.9291936	0.0000000
64-16	-15.00	-20.0708064	-9.9291936	0.0000000
48-32	-10.25	-15.3208064	-5.1791936	0.0000130
64-32	-5.25	-10.3208064	-0.1791936	0.0393966
64-48	5.00	-0.0708064	10.0708064	0.0548433

Cara membaca tabel tersebut adalah melihat apakah selang kepercayaan (lwr dan upper) memiliki tanda positif dan negatif. Jika ada dua tanda, berarti nol termasuk di selang kepercayaan, sehingga kita belum dapat menyatakan rata-rata perlakuan tersebut berbeda. Proses tersebut dapat divisualisasikan:

```
plot(tukey, las = 1, col = 'dark blue')
```

95% family-wise confidence level



Analisis di python dapat dimulai dengan melakukan reshape kepada data. Detail dari proses ini sudah dijelaskan sebelumnya:

```
import pandas as pd
```

```
RALData = pd.read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vT8sPRuEulqieCMFJK4
RALMeltpy=pd.melt(RALData, id_vars=['Dosis'], value_vars=[str(x) for x in range(1,9)])
RALMeltpy.rename(columns = {'variable':'Ulangan', 'value':'Produksi'}, inplace = True)
RALMeltpy.head()
```

	Dosis	Ulangan	Produksi
0	0	1	100
1	16	1	100
2	32	1	90
3	48	1	80
4	64	1	90

Lalu, statsmodels memiliki fungsi MultiComparison(respon, perlakuan):

```
import statsmodels.formula.api as smf
```

```
C:\Users\Acer\AppData\Local\R\win-library\4.2\reticulate\python\rpytools\loader.py:39: Future
module = _import(
C:\Users\Acer\AppData\Local\Programs\Python\PYTHON~2\lib\site-packages\statsmodels\compat\par
from pandas import Int64Index as NumericIndex
```

```
import statsmodels.stats.multicomp as multi
```

```
mcDosis = multi.MultiComparison(RALMeltpy['Produksi'],RALMeltpy['Dosis'])
Results = mcDosis.tukeyhsd()
print(Results)
```

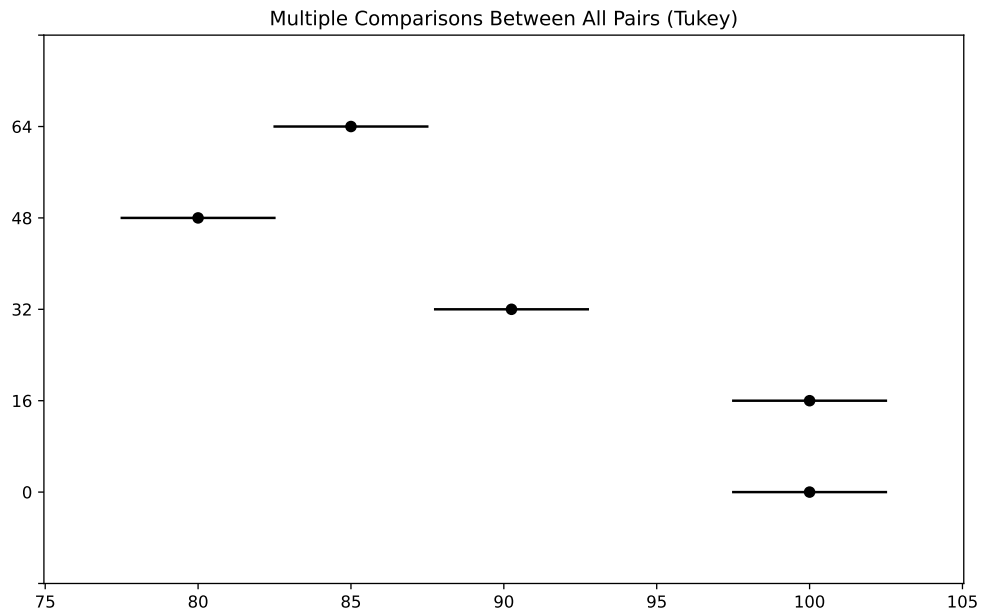
```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj  lower  upper  reject
=====
    0     16      0.0    0.9  -5.0708  5.0708  False
    0     32     -9.75  0.001 -14.8208 -4.6792   True
    0     48    -20.0  0.001 -25.0708 -14.9292   True
    0     64    -15.0  0.001 -20.0708 -9.9292   True
   16     32     -9.75  0.001 -14.8208 -4.6792   True
   16     48    -20.0  0.001 -25.0708 -14.9292   True
```

16	64	-15.0	0.001	-20.0708	-9.9292	True
32	48	-10.25	0.001	-15.3208	-5.1792	True
32	64	-5.25	0.0394	-10.3208	-0.1792	True
48	64	5.0	0.0549	-0.0708	10.0708	False

Hasil sama seperti R. Hasil juga dapat di-plot dengan:

```
import matplotlib.pyplot as plt

tukPlot=Results.plot_simultaneous()
plt.show()
```



Tingkat kepercayaan juga dapat diubah:

```
Results = mcDosis.tukeyhsd(alpha=0.01)
print(Results)
```

Multiple Comparison of Means - Tukey HSD, FWER=0.01

=====

group1	group2	meandiff	p-adj	lower	upper	reject
0	16	0.0	0.9	-6.2115	6.2115	False
0	32	-9.75	0.001	-15.9615	-3.5385	True
0	48	-20.0	0.001	-26.2115	-13.7885	True
0	64	-15.0	0.001	-21.2115	-8.7885	True
16	32	-9.75	0.001	-15.9615	-3.5385	True
16	48	-20.0	0.001	-26.2115	-13.7885	True
16	64	-15.0	0.001	-21.2115	-8.7885	True
32	48	-10.25	0.001	-16.4615	-4.0385	True
32	64	-5.25	0.0394	-11.4615	0.9615	False
48	64	5.0	0.0549	-1.2115	11.2115	False

3.3 Uji Duncan

Uji Duncan intinya mirip dengan uji Tukey, tetapi nilai pembanding di uji Duncan meningkat tergantung jarak dari perlakuan yang dibandingkan. Uji tersebut memiliki nilai kritis:

$$R_p = r_{\alpha; t; db_g} S_{\bar{y}}$$

$$S_{\bar{y}} = \sqrt{KTG/r}$$

Dan jika jumlah ulangan tak sama dapat didekati rata-rata harmonik juga. t adalah beda urutan perlakuan kedua kelompok. Misal membandingkan perlakuan terbesar (1) dan terkecil (4) dari 4 perlakuan akan menghasilkan $t=3$. Langkah uji mirip dengan Tukey, yaitu:

1. Urutkan rata-rata perlakuan dari yang terkecil sampai yang terbesar atau sebaliknya
2. Nilai awal $i=1$ (dari kiri) dan $j=1$ (dari kanan)
3. Hitung beda antara rata-rata perlakuan terkecil ke- i dengan terbesar ke- j kemudian bandingkan dengan nilai BNJ, jika beda rata-rata perlakuan lebih kecil lanjutkan ke langkah 5 (taktolak H_0) dan jika tidak lanjutkan ke langkah 4
4. Berikan $j = j+1$, jika $j < p$ kembali ke langkah 3
5. Buatlah garis mulai rata-rata perlakuan ke- i sampai ke perlakuan ke- j
6. Berikan $i = i+1$, jika $i < p$ kembali ke langkah 3
7. Stop

Langsung uji:

```
duncan <- duncan.test(aov_RAL, 'Dosis', alpha = 0.05) #Bisa Model aov atau lm
knitr::kable(duncan$duncan)
```

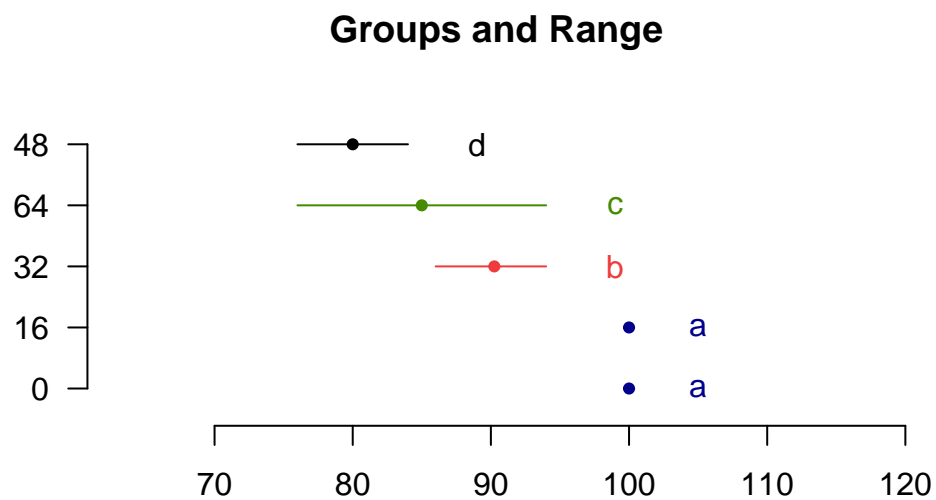
	Table	CriticalRange
2	2.871006	3.580545
3	3.018061	3.763944
4	3.113890	3.883455
5	3.182680	3.969247

```
knitr::kable(duncan$groups)
```

	Perkecambahan	groups
0	100.00	a
16	100.00	a
32	90.25	b
64	85.00	c
48	80.00	d

Dan dapat divisualisasikan:

```
plot(duncan, horiz = TRUE, las = 1)
```



Note bahwa Uji Tukey menyatakan dosis 64 tak beda dengan dosis 48, padahal uji Duncan dan LSD menyatakan dosis tersebut beda. Hal tersebut menunjukkan kurang sensitifnya uji Tukey.

3.4 Kontras

Kontras merupakan kombinasi linear dari rataaan perlakuan. Sebuah kombinasi linear tersusun atas seperangkat objek. Setiap objek dikalikan dengan suatu konstanta, dan hasil perkalian semua objek dan konstanta ditambahkan. Pada intinya, setiap rataan perlakuan akan dikalikan dengan suatu konstanta dan hasil perkalian tersebut akan ditambahkan. Bentuk umum kontras adalah:

$$H_0 : c_1\mu_1 + c_2\mu_2 + \cdots + c_t\mu_t = \sum_{i=1}^t c_i\mu_i = 0$$

$$H_1 : c_1\mu_1 + c_2\mu_2 + \cdots + c_t\mu_t = \sum_{i=1}^t c_i\mu_i \neq 0$$

Yang diduga dengan

$$C = \sum_{i=1}^t c_i y_i.$$

Dengan y_i adalah total di perlakuan ke-i. Kontras memiliki syarat:

$$\sum_{i=1}^t c_i = 0$$

Selain itu, kontras disebut *ortogonal* jika hasil suatu uji tidak memengaruhi hasil uji lain. Sebagai contoh, kontras:

$$H_0^{(1)} : \mu_1 = \mu_2$$

$$H_0^{(2)} : \mu_1 = \mu_3$$

$$H_0^{(3)} : \mu_2 = \mu_3$$

Tidak ortogonal. Bayangkan bahwa kita menolak H_0^3 , tetapi tak tolak hipotesis lainnya. Ini merupakan kontradiksi karena $\mu_1 = \mu_2$ dan $\mu_1 = \mu_3$ secara logis memberi implikasi $\mu_2 = \mu_3$. Namun, uji mengatakan sebaliknya.

Untuk melihat kenapa hal tersebut terjadi, kontras dapat dilihat lebih lanjut:

$$H_0^{(1)} : 1\mu_1 - 1\mu_2 + 0\mu_3 = 0$$

$$H_0^{(2)} : 1\mu_1 + 0\mu_2 - 1\mu_3 = 0$$

$$H_0^{(3)} : 0\mu_1 + 1\mu_2 - 1\mu_3 = 0$$

Dapat dengan mudah diverifikasi bahwa semua $\sum_{i=1}^t c_i$ adalah nol. Namun, jika kita kalikan kontras pertama (c_i) dan kontras kedua (d_i) serta melihat hasilnya, maka kita akan temukan $\sum_{i=1}^t c_i d_i = 1 + 0 + 0 = 1 \neq 0$. Bandingkan dengan contoh ini:

Sebuah percobaan menguji efek pemberian nitrogen pada pertumbuhan rumput. Perlakuan dalam percobaan tersebut ada empat, yaitu tidak diberi nitrogen sebagai kontrol (1), diberi nitrogen di musim gugur (2), diberi nitrogen di musim semi (3), dan diberikan pada kedua musim (4). Ingin diketahui:

1. Apakah terdapat perbedaan pertumbuhan rumput dari yang tidak diberi nitrogen dengan diberi nitrogen?
2. Apakah ada perbedaan pertumbuhan rumput jika diberikan nitrogen pada saat musim gugur dengan musim semi?
3. Apakah ada perbedaan pertumbuhan rumput jika diberikan pada saat musim gugur atau musim semi dengan diberikan pada kedua musim tersebut?

Kontras tersebut dapat disusun:

$$H_0^{(1)} : 3\mu_1 - 1\mu_2 - 1\mu_3 - 1\mu_4 = 0$$

$$H_0^{(2)} : 0\mu_1 + 1\mu_2 - 1\mu_3 - 0\mu_4 = 0$$

$$H_0^{(3)} : 0\mu_1 + 1\mu_2 + 1\mu_3 - 2\mu_4 = 0$$

Alasan penyusunan kontras sebagai berikut dapat ditelusuri dari beberapa prinsip:

1. Di sebuah kontras, perlakuan yang tidak ingin dibandingkan mendapatkan $c_i = 2$. Misal, di kontras kedua hanya dibandingkan musim gugur (μ_2) dan musim semi (μ_3), sehingga koefisien di tanpa nitrogen dan kedua musim (μ_1, μ_4) nol. Begitu juga di kontras ketiga, perlakuan tanpa nitrogen tidak ingin dibandingkan sehingga koefisiennya nol.
2. Pastikan bobot tiap perlakuan sama. Suatu kontras pada dasarnya membagi perlakuan menjadi dua kelompok yang hendak dibandingkan. Namun, agar perbandingan tersebut adil, bobot perlakuan harus sama. Misal, di hipotesis pertama, hanya ada 1 perlakuan tanpa nitrogen sedangkan ada 3 perlakuan dengan nitrogen. Jika kita membandingkan rata-rata 1 perlakuan tanpa nitrogen dengan hasil tambah *tiga* perlakuan dengan nitrogen, pasti tidak adil. Anggap saja $\mu_1 = \mu_2 = \mu_3 = \mu_4$, seharusnya kontras menunjukkan sisi kiri sama dengan sisi kanan. Namun, jika kita melakukan pertambahan secara naif, akan muncul perbandingan $\mu_1 \neq 3\mu_1$. Oleh karena itu μ_1 harus kita boboti 3 agar perbandingannya adil. Dalam kasus H_0 benar tadi, pembobotan μ_1 dengan 3 menghasilkan $3\mu_1 = 3\mu_1$.
3. Pastikan tanda antara kelompok berbeda. Satu kelompok akan memiliki tanda positif dan satu kelompok memiliki tanda negatif. Ini dikarenakan jumlah koefisien nol sehingga perlu tanda positif dan negatif untuk menghasilkan angka tersebut.

Sekarang, dapat dipikirkan secara logis implikasi dari tiap uji:

$$H_0^{(1)}: \mu_1 = \frac{\mu_2 + \mu_3 + \mu_4}{3}$$

$$H_0^{(2)}: \mu_2 = \mu_3$$

$$H_0^{(3)}: \frac{\mu_2 + \mu_3}{2} = \mu_4$$

Tidak ada hipotesis yang saling berkontradiksi. Jika μ_2 tidak sama dengan μ_3 , misal, hipotesis pertama dan ketiga masih dapat benar ataupun salah. Jika dijabarkan dalam perkataan, pengujian mengenai efek dari pemberian nitrogen dibandingkan dengan tidak sama sekali tidak memberi informasi mengenai efek pemberian nitrogen di dua musim dibandingkan dengan satu musim. Begitu juga, kedua uji tersebut tidak memberi informasi mengenai efek perbandingan nitrogen di musim semi atau gugur.

Secara formal, dapat kita lihat bahwa $\sum_{i=1}^t c_i d_i = 0$ untuk tiap kombinasi kontras. Dari aljabar linear, jika \mathbf{c} dan \mathbf{d} merupakan vektor, ini sama dengan hasil kali transpose \mathbf{c} dengan \mathbf{d} . Maka, buktikan di R:

```
kontras1<-c(3,-1,-1,-1)
kontras2<-c(0,1,-1,0)
kontras3<-c(0,1,1,-2)
```

```
#kali vektor
t(kontras1)%*%kontras2
```

```
      [,1]
[1,]      0
```

```
t(kontras1)%*%kontras3
```

```
      [,1]
[1,]      0
```

```
t(kontras2)%*%kontras3
```

```
      [,1]
[1,]      0
```

Oleh karena itu, kontras akan orthogonal jika $\sum_{i=1}^t c_i d_i = 0$. Keortogonalan berarti hasil satu uji bebas dari uji lain, sehingga jumlah kuadrat perlakuan dapat dipartisi.

3.4.1 Menyusun kontras ortogonal

Misalkan ada contoh:

Ingin dibandingkan 5 populasi dari data. Sebelumnya, kita memiliki alasan untuk percaya bahwa populasi 2 dan 3 paling dekat. Populasi 4 dan 5 juga dekat, tetapi tidak sedekat populasi 2 dan 3. Populasi 1 lebih dekat ke populasi 2 dan 3 daripada populasi 4 dan 5.

Kedekatan tersebut digambarkan dalam diagram:

Dari grafik dan soal tersebut, muncul empat pertanyaan: 1. Apakah populasi 2 dan 3 (yang dianggap sekelompok) beda? (D) 2. Apakah populasi 4 dan 5 beda? (C) 3. Apakah populasi 1 beda dengan populasi 2, 3? (B) 4. Apakah populasi 1, 2, 3, beda dengan populasi 4 dan 5? (A)

Oleh karena itu, dapat dibentuk kontras:

Kontras	1	2	3	4	5
A	2	2	2	-3	-3
B	2	-1	-1	0	0
C	0	0	0	1	-1
D	0	1	-1	0	0

Keortogonalan kontras tersebut dapat dicek melalui matriks ragam peragam:

```
A<-c(2,2,2,-3,-3)
B<-c(2,-1,-1,0,0)
C<-c(0,0,0,1,-1)
D<-c(0,1,-1,0,0)

contrasts<-cbind(A,B,C,D)
cov(contrasts)
```

```
      A    B    C    D
A 7.5 0.0 0.0 0.0
B 0.0 1.5 0.0 0.0
C 0.0 0.0 0.5 0.0
D 0.0 0.0 0.0 0.5
```

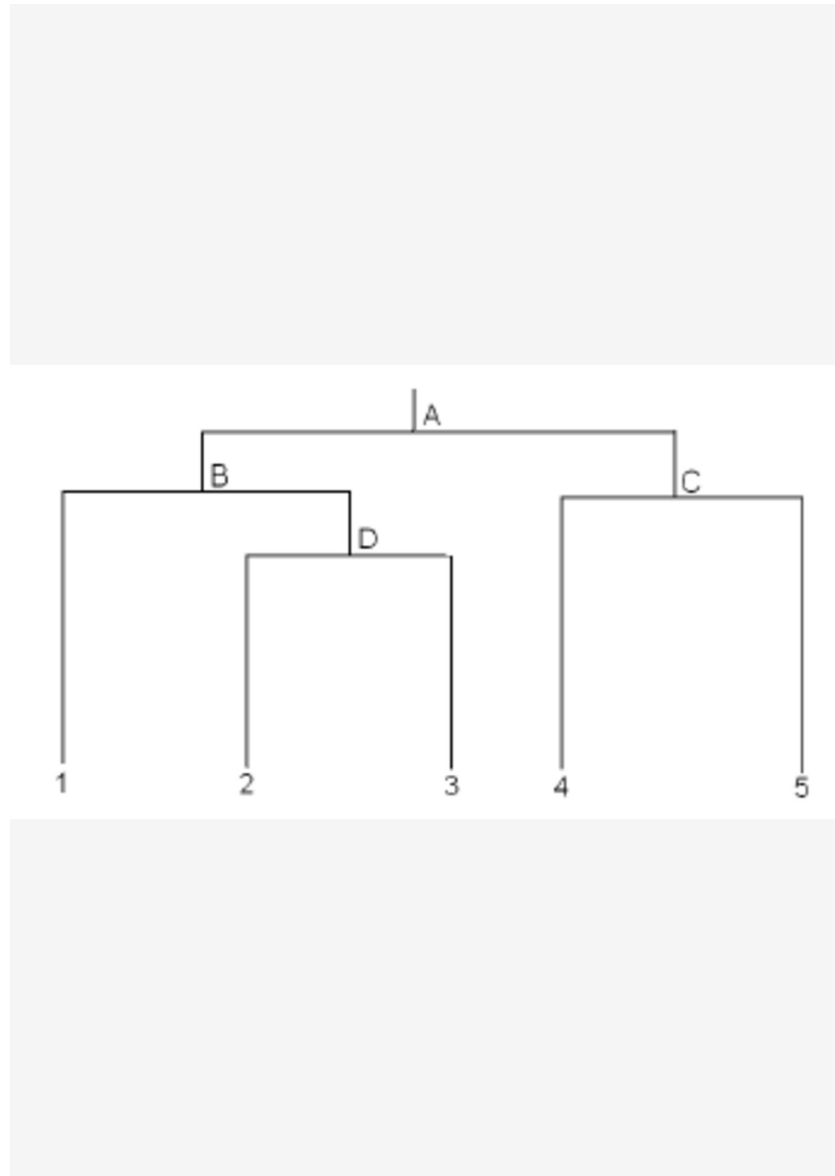


Figure 3.1: Diagram pengelompokan populasi. Sumber: Department of Statistics, Pennsylvania State University (online.stat.psu.edu).

Jika matriks diagonal, ini berarti koragam nol dan kontras saling orthogonal. Secara logis, hasil tiap uji tidak memberi informasi mengenai hasil uji lainnya.

Contoh lain untuk mengilustrasikan kontras orthogonal adalah sebagai berikut

Efektivitas suatu obat hendak dicek. Perlakuan 1 merupakan obat A berdosis rendah, perlakuan 2 merupakan obat A berdosis tinggi, perlakuan 3 merupakan obat B berdosis rendah, dan perlakuan 4 merupakan obat B berdosis tinggi.

Tentu, terlihat beberapa arah perbandingan:

1. Obat - apakah obat A dan B beda?
2. Dosis - apakah dosis tinggi dan rendah beda?
3. Interaksi - apakah efek obat bergantung pada dosis?

Sehingga dibuat kontras:

Kontras	1	2	3	4
Obat	1	1	-1	-1
Dosis	1	-1	1	-1
Interaksi	1	-1	-1	1

Cek ortogonalitas

```
Obat<-c(1,1,-1,-1)
Dosis<-c(1,-1,1,-1)
Interaksi<-c(1,-1,-1,1)

contrasts2<-cbind(Obat,Dosis,Interaksi)
cov(contrasts2)
```

```
      Obat    Dosis Interaksi
Obat    1.333333 0.000000  0.000000
Dosis    0.000000 1.333333  0.000000
Interaksi 0.000000 0.000000  1.333333
```

Alasan penyusunan kontras obat dan dosis seperti di tabel cukup masuk akal. Bandingkan obat A dan B atau kelompok dosis tinggi dan dosis rendah. Penyusunan kontras interaksi sebenarnya sama saja dengan mengalikan efek-efek sebelumnya. Telah disusun obat A bertanda positif (+) dan obat B bertanda negatif (-). Dosis rendah bertanda positif (+) dan dosis tinggi bertanda negatif. Maka, kalikan saja tanda tersebut.

Sebagai contoh lain, misal diberi tabel rata-rata perlakuan sebagai berikut:

Perlakuan	A	B	C	D
Rata-rata	12.0	12.3	10.8	6.8

Dapat dilihat bahwa D tampak sangat berbeda dari perlakuan lain. A dan B tampak sangat mirip. C agak dekat ke A dan B. Oleh karena itu, susun kontras:

Kontras	A	B	C	D
D vs ABC	1	1	1	-3
c vs AB	1	1	-2	0
A vs B	1	-1	0	0

Orthogonalitas kontras tersebut dapat diuji:

```
DvABC<-c(1,1,1,-3)
CvAB<-c(1,1,-2,0)
AvB<-c(1,-1,0,0)

contrasts3<-cbind(DvABC,CvAB,AvB)
cov(contrasts3)
```

```
      DvABC CvAB      AvB
DvABC      4    0 0.0000000
CvAB       0    2 0.0000000
AvB        0    0 0.6666667
```

3.4.2 Menghitung Kontras

Setelah mengetahui cara membuat kontras, bagaimana perhitungannya? Untuk t-1 kontras orthogonal, dapat dilakukan uji F, dengan statistik uji sebagai berikut:

$$JKC = KTC = \frac{\left(\sum_{i=1}^t c_i Y_{i.}\right)^2}{r \sum_{i=1}^t c_i^2}$$

$$F = KTC/KTG = F_{\alpha,1,db_g}$$

Dapat dipraktekkan di R:

```
library(googleSheets4)
dataKontras<-read_sheet("https://docs.google.com/spreadsheets/d/19l0rOm5Vb7ATet8tR6Xq5NEOn
```

v Reading from "Kontras".

v Range 'Sheet1'.

```
knitr::kable(dataKontras)
```

Perlakuan	Kelompok 1	Kelompok 2	Kelompok 3
Kontrol	9.9	12.3	11.4
Fall	11.4	12.9	12.7
Spring	12.1	13.4	12.9
Split	10.1	12.2	11.9

Tentu, data harus di-melt:

```
library(reshape2)
KontrasMelted<-melt(dataKontras,
                      id.vars=c("Perlakuan"),
                      measured.vars=c("Kelompok 1", "Kelompok 2","Kelompok 3"),
                      value.name="Pertumbuhan")
colnames(KontrasMelted)[2]<-"Kelompok"
```

Lalu, pastikan perlakuan dan kelompok adalah faktor dan buat kontras:

```
#faktor
KontrasMelted$Perlakuan<-as.factor(KontrasMelted$Perlakuan)
KontrasMelted$Kelompok<-as.factor(KontrasMelted$Kelompok)

levels(KontrasMelted$Perlakuan)
```

```
[1] "Fall"      "Kontrol"  "Split"    "Spring"
```

Kita sudah mengetahui urutan faktor di R. Oleh karena itu, buat kontras:

```
#buat kontras
contrasts(KontrasMelted$Perlakuan) <- cbind(c(1, -3, 1, 1), c(1, 0, 0, -1),
                                              c(1, 0, -2, 1))
contrasts(KontrasMelted$Perlakuan)
```

	[,1]	[,2]	[,3]
Fall	1	1	1
Kontrol	-3	0	0
Split	1	0	-2
Spring	1	-1	1

Lakukan ANOVA dan partisi:

```
aovKontras<-aov(Pertumbuhan~Perlakuan+Kelompok,KontrasMelted)
summary.aov(aovKontras, split = list (Perlakuan = list('K vs FSpSpl' = 1, 'F vs Sp' = 2, 'F vs K' = 3),
Kelompok = list('Musim 1' = 1, 'Musim 2' = 2)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Perlakuan	3	5.200	1.733	19.439	0.001713 **
Perlakuan: K vs FSpSpl	1	2.151	2.151	24.125	0.002679 **
Perlakuan: F vs Sp	1	0.327	0.327	3.664	0.104123
Perlakuan: FSp vs Spl	1	2.722	2.722	30.530	0.001480 **
Kelompok	2	7.172	3.586	40.215	0.000335 ***
Residuals	6	0.535	0.089		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Interpretasi dari partisi tersebut sesuai dengan hipotesis yang diuji. Jika menggunakan $\alpha = 5\%$, dinyatakan bahwa rata-rata kontrol dengan perlakuan lainnya berbeda signifikan, begitu juga rata-rata 2 musim dengan 1 musim. Namun, belum ditemukan bukti cukup untuk menyatakan rata-rata musim semi dan gugur berbeda signifikan.

3.5 Polinomial Orthogonal

Polinomial orthogonal dilakukan untuk menguji tren pengaruh perlakuan terhadap respon jika perlakuan bersifat kuantitatif dan berjarak sama. Model polinomial orthogonal dapat berbentuk linear, kuadrat, kubik, kuartik, dan lain-lain:

$$\text{Linear: } Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

$$\text{Kuadrat: } Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \varepsilon_i$$

$$\text{Kubik: } Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 X_i^3 + \varepsilon_i$$

Dengan bentuk umum polinomial ordo ke-N adalah:

$$Y = \alpha_0 P_0(X) + \alpha_1 P_1(X) + \alpha_2 P_2(X) + \dots + \alpha_n P_n(X) + \varepsilon_i$$

Dengan:

$$\begin{aligned}
 P_0 &= 1 \\
 P_1 &= \lambda_1 \left[\frac{X - \bar{X}}{d} \right] \\
 P_2 &= \lambda_2 \left[\left(\frac{X - \bar{X}}{d} \right)^2 - \left(\frac{a^2 - 1}{12} \right) \right] \\
 P_{n+1}(X) &= \lambda_{n+1} \left[P_1(X)P_n(X) - \frac{n^2(a^2 - n^2)}{4(4n^2 - 1)} P_{n-1}(X) \right]
 \end{aligned}$$

Dengan:

1. a : banyaknya taraf faktor
2. d : jarak antar faktor
3. n : polinomial ordo ke- n

Tabel kontras bagi polinomial orthogonal untuk a jumlah faktor dapat dihasilkan dengan `contr.poly(a)`:

```
knitr::kable(contr.poly(5), col.names=c("Linear",
                                           "Kuadratik",
                                           "Kubik",
                                           "Kuartik"))
```

Linear	Kuadratik	Kubik	Kuartik
-0.6324555	0.5345225	-0.3162278	0.1195229
-0.3162278	-0.2672612	0.6324555	-0.4780914
0.0000000	-0.5345225	0.0000000	0.7171372
0.3162278	-0.2672612	-0.6324555	-0.4780914
0.6324555	0.5345225	0.3162278	0.1195229

Untuk memperlihatkan cara kerja polinomial orthogonal, dapat dibuat suatu data yang memiliki taraf-taraf faktor kuantitatif:

```
kerapatan <- factor(rep(c(10, 20, 30, 40, 50), each = 3))
hasil <- c(12.2, 11.4, 12.4,
           16.0, 15.5, 16.5,
           18.6, 20.2, 18.2,
           17.6, 19.3, 17.1,
           18.0, 16.4, 16.6)
data2 <- data.frame(kerapatan, hasil)
str(data2)
```

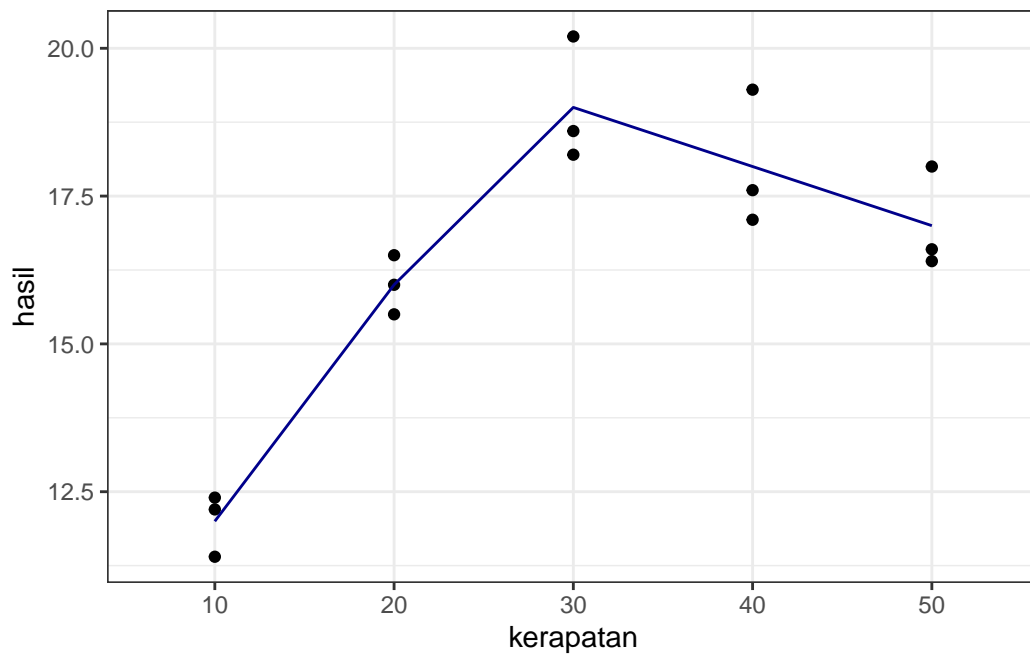
```
'data.frame': 15 obs. of 2 variables:
 $ kepadatan: Factor w/ 5 levels "10","20","30",...: 1 1 1 2 2 2 3 3 3 4 ...
 $ hasil    : num 12.2 11.4 12.4 16 15.5 16.5 18.6 20.2 18.2 17.6 ...
```

Sebelum menganalisis menggunakan polinomial, plot data sebagai cara eksploratif:

```
library(ggplot2)

ggplot(data2,aes(x=kepadatan,y=hasil))+
  geom_point()+
  stat_summary(fun.y=mean, colour="darkblue", geom="line", aes(group = 1))+
  theme_bw()
```

Warning: `fun.y` is deprecated. Use `fun` instead.



Tampaknya pengaruh kepadatan berbentuk kuadratik karena plot menarik, mencapai puncak, lalu menurun. Lalu, definisikan kontras dan lakukan ANOVA.

```
contrasts(data2$kepadatan) <- contr.poly(5)
```

```
anova2 <- aov(hasil ~ kerapatan, data = data2)
summary.aov(anova2, split = list(kerapatan = list('Linear' = 1, 'Kuadratik' = 2,
                                                    'Kubik' = 3, 'Kuartik' = 4)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
kerapatan	4	87.60	21.90	29.278	1.69e-05 ***
kerapatan: Linear	1	43.20	43.20	57.754	1.84e-05 ***
kerapatan: Kuadratik	1	42.00	42.00	56.150	2.08e-05 ***
kerapatan: Kubik	1	0.30	0.30	0.401	0.541
kerapatan: Kuartik	1	2.10	2.10	2.807	0.125
Residuals	10	7.48	0.75		

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Nampaknya pengaruh kerapatan pada hasil berupa kuadratik, yang dapat dimodelkan sebagai:

```
summary(lm(hasil~kerapatan,data2))
```

Call:

```
lm(formula = hasil ~ kerapatan, data = data2)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.90	-0.55	-0.40	0.45	1.30

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	16.4000	0.2233	73.441	5.35e-15 ***
kerapatan.L	3.7947	0.4993	7.600	1.84e-05 ***
kerapatan.Q	-3.7417	0.4993	-7.493	2.08e-05 ***
kerapatan.C	0.3162	0.4993	0.633	0.541
kerapatan^4	0.8367	0.4993	1.676	0.125

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8649 on 10 degrees of freedom

Multiple R-squared: 0.9213, Adjusted R-squared: 0.8899

F-statistic: 29.28 on 4 and 10 DF, p-value: 1.69e-05

$$Y = 16.400 + 3.7947X_i - 3.7417X_i^2$$

Kontras juga dapat diinput secara manual.

4 Uji Asumsi

Prosedur ANOVA mengasumsikan beberapa hal mengenai struktur model dan struktur galat. Jika asumsi tersebut tidak terpenuhi, hasil dari ANOVA mungkin saja tidak valid. Oleh karena itu, pengujian asumsi sangat penting untuk diketahui.

4.1 Galat percobaan menyebar normal

4.1.1 Prosedur Eksploratif

Secara eksploratif dalam mendeteksi kenormalan pada galat dapat menggunakan plot peluang normal (Normal probability (Q-Q) plot) dan histogram sisaan.

Metode grafis plot peluang normal (Normal probability (Q-Q) plot) memiliki sumbu vertikal plot (nilai data yang berurut) dan sumbu horizontal plot (nilai peluang normal). Untuk mendeteksi normalitas pada residual, maka pada sumbu vertikal plot NPP merupakan residual yang terurut dan pada sumbu horizontal merupakan skor peluang normal yang terurut dari residual. Jika pada NPP diperoleh garis yang lurus atau mendekati/hampir lurus maka data residual tersebut menyebar normal. Garis tersebut tidak akan betul-betul lurus sempurna karena adanya keragaman contoh. Jika dibuat plot NPP untuk data tidak normal, plot akan cenderung melengkung, dan bentuk kelengkungan tersebut dapat menjadi informasi sejauh mana penyimpangan ketidaknormalannya.

```
library(google sheets4)
DataRAKL<-read_sheet("https://docs.google.com/spreadsheets/d/1uFi_Njharot8G8-DvHWPhNLLqJ72
```

```
! Using an auto-discovered, cached token.
```

```
To suppress this message, modify your code or options to clearly consent to
the use of a cached token.
```

```
See gargle's "Non-interactive auth" vignette for more details:
```

```
<https://gargle.r-lib.org/articles/non-interactive-auth.html>
```


i The googlesheets4 package is using a cached token for
'muh.ammarsahab@apps.ipb.ac.id'.

v Reading from "RAKL_P2 Rancob".

v Range 'Sheet1'.

```
DataRAKL$Produksi<-as.numeric(DataRAKL$Produksi)

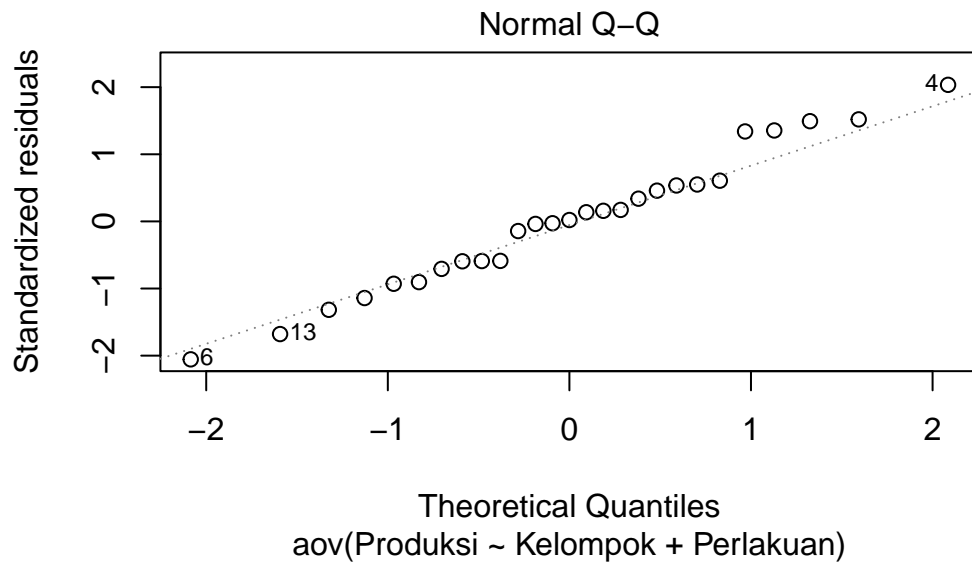
#pastikan perlakuan dan kelompok berbentuk faktor
DataRAKL$Kelompok<-as.factor(DataRAKL$Kelompok)
DataRAKL$Perlakuan<-as.factor(DataRAKL$Perlakuan)

aovRAKL<-aov(Produksi~Kelompok+Perlakuan,data=DataRAKL) #prosedur anova
broom::tidy(aovRAKL) |> knitr::kable()
```

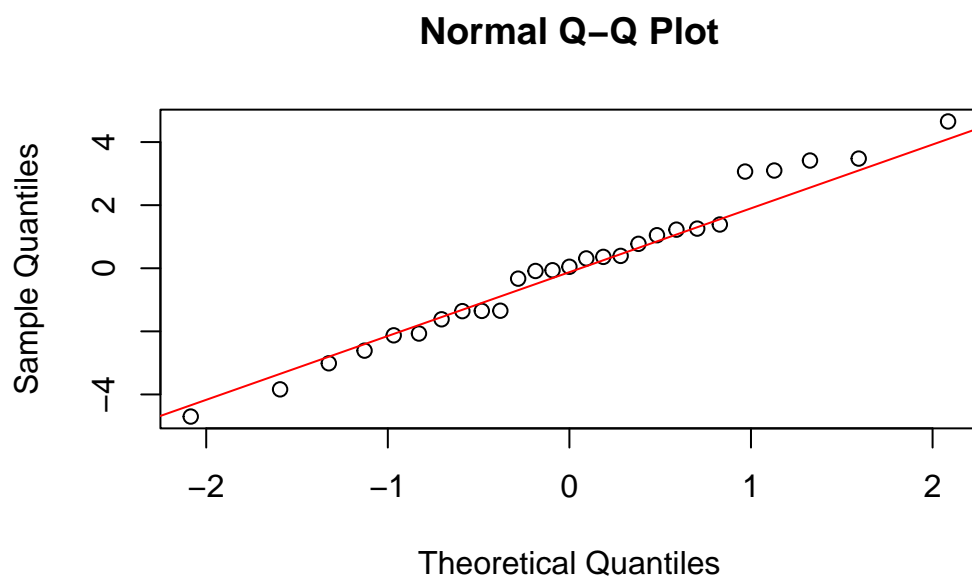
term	df	sumsq	meansq	statistic	p.value
Kelompok	2	39.2107	19.605348	2.220614	0.1408977
Perlakuan	8	586.0398	73.254970	8.297279	0.0001896
Residuals	16	141.2607	8.828794	NA	NA

Pertama, plot Q-Q:

```
# Plot Peluang Normal (Q-Q)
plot(aovRAKL, which = 2)
```

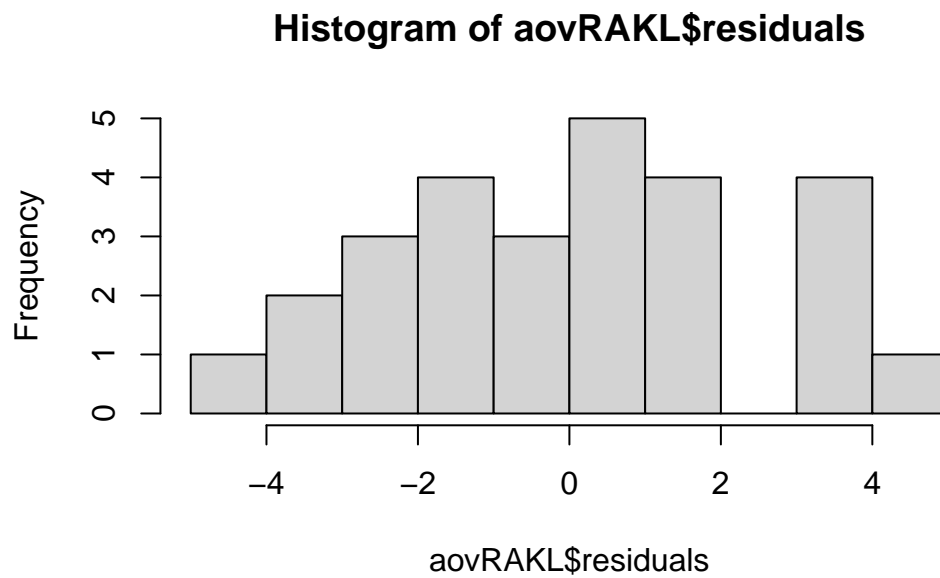


```
#atau#
qqnorm(aovRAKL$residuals); qqline(aovRAKL$residuals, col = 'red')
```

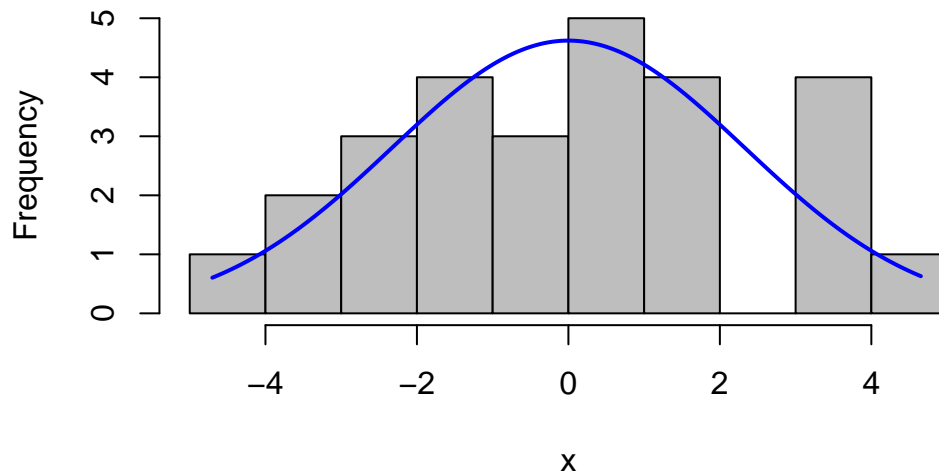


Lalu, histogram:

```
# Histogram Sisaan  
hist(aovRAKL$residuals,breaks = 10)  
#atau#  
library(rcompanion)
```



```
plotNormalHistogram(aovRAKL$residuals, breaks = 10)
```



Dari plot di atas dapat terlihat bahwa semua titik cenderung berada kira-kira di sekitar sepanjang garis diagonal tetapi ada satu titik tidak dekat dengan garis dan histogram membentuk seperti kurva sebaran normal sehingga asumsi galat percobaan menyebar normal terpenuhi. Metode qq-plot dan histogram residual di python sebagai berikut. Pertama, lakukan ANOVA:

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
```

```
C:\Users\Acer\AppData\Local\R\win-library\4.2\reticulate\python\rpytools\loader.py:39: FutureWarning:
  module = _import(
```

```
C:\Users\Acer\AppData\Local\Programs\Python\PYTHON~2\lib\site-packages\statsmodels\compat\pandas.py:10:
  from pandas import Int64Index as NumericIndex
```

```
from statsmodels.formula.api import ols
import seaborn as sns
from matplotlib import pyplot as plt
```

```
RAKLpy=pd.read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vTi9y0_OJZ5pHMmPRlPqPx")
```

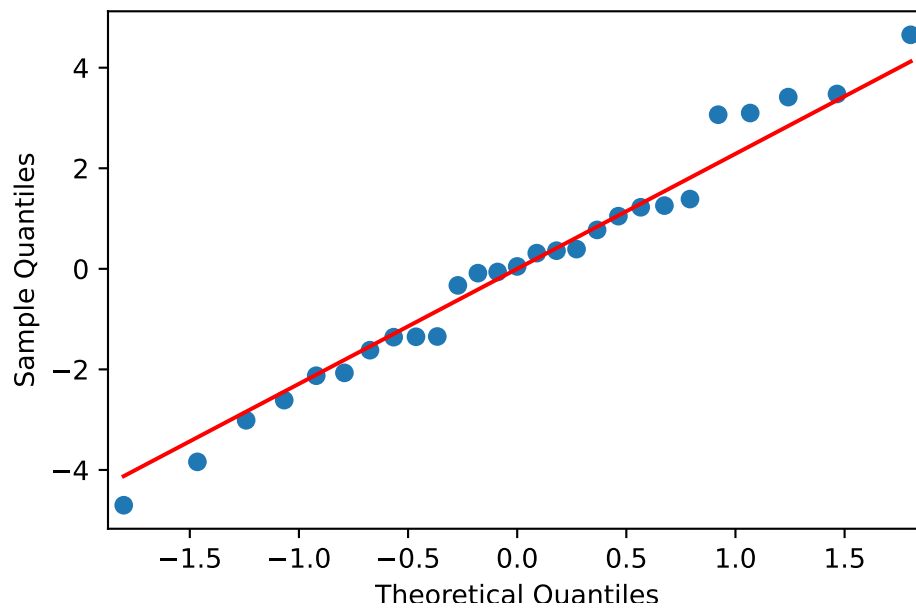
```
RAKL_lm= ols('Produksi ~ C(Perlakuan)+C(Kelompok)',data=RAKLpy).fit() #C=peubah kategorik
table = sm.stats.anova_lm(RAKL_lm, typ=2) # Type 2 ANOVA DataFrame
print(table)
```

	sum_sq	df	F	PR(>F)
C(Perlakuan)	586.039763	8.0	8.297279	0.000190
C(Kelompok)	39.210696	2.0	2.220615	0.140898
Residual	141.260704	16.0	NaN	NaN

Buat plot:

```
import statsmodels.api as sm
import matplotlib.pyplot as plt

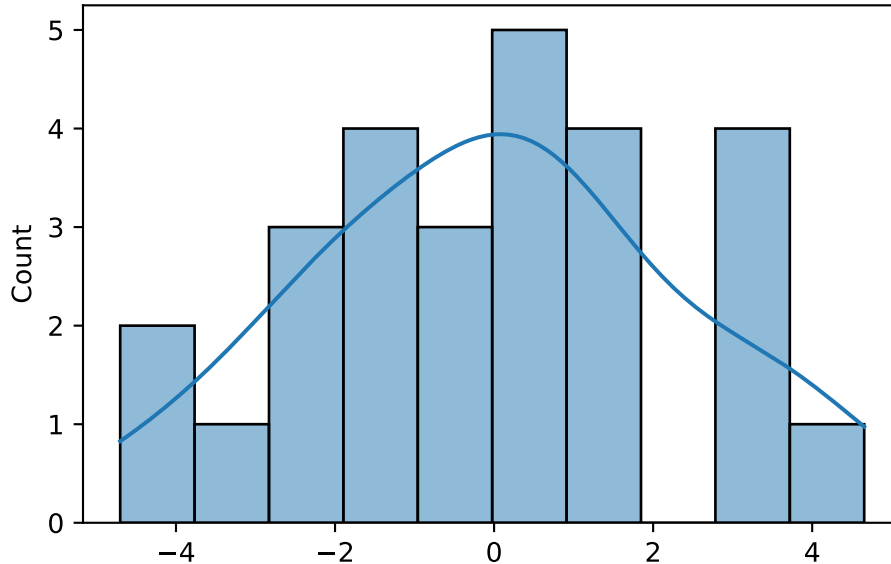
#create Q-Q plot with 45-degree line added to plot
fig = sm.qqplot(RAKL_lm.resid, line='s')
plt.show()
```



Dan untuk histogram:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.clf() #clear plot sebelumnya
h=sns.histplot(RAKL_lm.resid,bins=10, kde=True)
plt.show()
```



4.1.2 Uji formal untuk galat percobaan menyebar normal

secara formal dapat menggunakan beberapa uji tertentu diantaranya adalah Uji Kolmogorov-Smirnov, Uji Shapiro-Wilk, Uji Lilliefors, Uji Jarque-Bera, Uji Anderson-Darling, dsb.

Semua uji tersebut memiliki hipotesis:

H_0 : Galat menyebar normal

H_1 : Galat tidak menyebar normal

Uji Shapiro Wilk memiliki statistik uji sebagai berikut:

$$W = \frac{\left[\sum_{i=1}^n a_i x_{(i)} \right]^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$x_{(i)}$ adalah statistik tataan ke-i - angka terkecil ke-i dari sampel. a_i berasal dari suatu rumus berisi mean dan matriks peragam X . Implementasi di R:

```
shapiro.test(x = aovRAKL$residuals)
```

Shapiro-Wilk normality test

```
data: aovRAKL$residuals  
W = 0.98096, p-value = 0.8833
```

Dan implementasi python:

```
from scipy.stats import shapiro  
  
result = (shapiro(RAKL_lm.resid))  
  
print(f"S-W statistic: {result[0]}")
```

```
S-W statistic: 0.9809552431106567
```

```
print(f"p-value: {result[1]}")
```

```
p-value: 0.8832639455795288
```

Dari keluaran uji Shapiro-Wilk dapat dilihat nilai $p\text{-value} = 0.8832 > \alpha = 0.05$ maka tak tolak H_0 , artinya tidak cukup bukti untuk mengatakan bahwa galat tidak menyebar normal sehingga asumsi galat percobaan menyebar normal terpenuhi.

Uji Kolmogorov-Smirnov memiliki statistik uji:

$$D_n = \sup_x |F_n(x) - F(x)|$$

Atau, jarak terbesar antara fungsi kumulatif dari sampel dan fungsi kumulatif dari sebaran normal (dalam kasus menguji kenormalan). Fungsi kumulatif dapat disesuaikan untuk sebaran yang berbeda jika ingin menguji kesesuaian dengan sebaran lain. Visualisasi dari proses ini ditunjukkan di [dokumen ini](#).

```
# Uji Kolmogorov-Smirnov
ks.test(aovRAKL$residuals, pnorm, mean(aovRAKL$residuals), sd(aovRAKL$residuals))
```

Exact one-sample Kolmogorov-Smirnov test

```
data: aovRAKL$residuals
D = 0.090934, p-value = 0.964
alternative hypothesis: two-sided
```

Dan di python:

```
from scipy.stats import kstest
import numpy as np

result = kstest(RAKL_lm.resid, cdf="norm", args=(np.mean(RAKL_lm.resid), np.std(RAKL_lm.resid)))

print(f"K-S statistic: {result[0]}")
```

K-S statistic: 0.09507410791382365

```
print(f"p-value: {result[1]}")
```

p-value: 0.9483937411797532

Dari keluaran diatas dapat dilihat nilai $p\text{-value} = 0.964$ (R) $>$ $\alpha = 0.05$ maka Tak Tolak H_0 , artinya tidak cukup bukti untuk mengatakan bahwa galat tidak menyebar normal sehingga dengan uji Kolmogorov-Smirnov asumsi galat percobaan menyebar normal juga terpenuhi.

Uji Lilliefors pada dasarnya merupakan modifikasi dari uji Kolmogorov Smirnov. Uji tersebut dapat dilakukan tanpa menspesifikasi mean dan simpangan baku dari sebaran normal yang hendak diuji, serta memberikan koreksi pada nilai Kolmogorov Smirnov. Implementasi di R:

```
# Uji Lilliefors
# Hipotesis: H0 : galat menyebar normal
#           H1 : galat tidak menyebar normal
library(nortest)
lillie.test(aovRAKL$residuals)
```


Lilliefors (Kolmogorov-Smirnov) normality test

```
data: aovRAKL$residuals  
D = 0.090934, p-value = 0.8199
```

```
from statsmodels.stats.diagnostic import lilliefors  
  
D,p=lilliefors(RAKL_lm.resid,dist='norm',pvalmethod='table')  
print("Statistik uji: ",D)
```

Statistik uji: 0.09093377335390318

```
print("p-value: ",p)
```

p-value: 0.8117606135583574

Dari keluaran di atas dapat dilihat nilai $p\text{-value} = 0.8199 > \alpha = 0.05$ maka Tak Tolak H_0 , artinya tidak cukup bukti untuk mengatakan bahwa galat tidak menyebar normal sehingga dengan uji Lilliefors asumsi galat percobaan menyebar normal juga terpenuhi.

Uji Jarque-Bera membandingkan skewness dan kurtosis data dengan sebaran normal. Ini dapat dilihat dari statistik uji:

$$JB = \frac{n}{6} \times \left(S^2 + \frac{(K-3)^2}{4} \right)$$

n adalah jumlah sampel, S adalah skewness, dan K adalah kurtosis. Implementasi di R sebagai berikut:

```
# Uji Jarque-Bera  
# Hipotesis: H0 : galat menyebar normal  
#           H1 : galat tidak menyebar normal  
library(tseries)  
tseries::jarque.bera.test(aovRAKL$residuals)
```

Jarque Bera Test

```
data: aovRAKL$residuals  
X-squared = 0.31309, df = 2, p-value = 0.8551
```

Dan implementasi python:

```
from scipy.stats import jarque_bera

result = (jarque_bera(RAKL_lm.resid))

print("JB statistic:", result[0])
```

JB statistic: 0.3130851694293787

```
print("p-value:", result[1])
```

p-value: 0.8550951029465681

Dari keluaran diatas dapat dilihat nilai $p\text{-value} = 0.8551 > \alpha = 0.05$ maka Tak Tolak H_0 , artinya tidak cukup bukti untuk mengatakan bahwa galat tidak menyebar normal sehingga dengan uji Jarque-Bera asumsi galat percobaan menyebar normal juga terpenuhi.

Uji Anderson-Darling merupakan modifikasi dari uji Kolmogorov Smirnov:

$$A^2 = n \int_{-\infty}^{\infty} \left(F_n(x) - F(x) \right)^2 w(x) dF(x)$$

Statistik uji tersebut masih merupakan beda antara fungsi kumulatif, tetapi beda yang diambil tidak hanya beda terbesar. Selain itu, ada pembobotan:

$$w(x) = \left[F(x)(1 - F(x)) \right]^{-1}$$

Sehingga bobot tersebut akan lebih besar di ujung sebaran. Implementasi di R sebagai berikut:

```
# Uji Anderson-Darling
# Hipotesis: H0 : galat menyebar normal
#           H1 : galat tidak menyebar normal
library(nortest)
ad.test(aovRAKL$residuals)
```

Anderson-Darling normality test

```
data: aovRAKL$residuals  
A = 0.23022, p-value = 0.7851
```

Sedangkan di python:

```
from scipy.stats import anderson  
  
result = (anderson(RAKL_lm.resid, dist='norm'))  
  
print("A-D statistic:", result[0])
```

A-D statistic: 0.2302183152399664

```
print("Critical values:", result[1])
```

Critical values: [0.517 0.589 0.707 0.824 0.98]

```
print("Significance levels:", result[2])
```

Significance levels: [15. 10. 5. 2.5 1.]

Dari keluaran diatas dapat dilihat nilai $p\text{-value} = 0.7851 > /alpha = 0.05$ maka Tak Tolak H_0 , artinya tidak cukup bukti untuk mengatakan bahwa galat tidak menyebar normal sehingga dengan uji Anderson-Darling asumsi galat percobaan menyebar normal juga terpenuhi. Interpretasi output Python adalah dengan membandingkan statistik uji dengan nilai kritis di taraf tertentu. Misal, karena $0.2302 < 0.517$, H_0 tak ditolak di $\alpha = 15\%$. Jika nilai statistik uji lebih besar dari nilai kritis, H_0 dapat ditolak.

4.1.3 Konsekuensi dari ketidaknormalan pada galat.

1. Kemenjuluran mempengaruhi selang kepercayaan

Kemenjuluran digunakan untuk mengukur ketidaksimetrisan data dan kurtosis digunakan untuk mengukur panjang ekor pada data. Baik kemenjuluran dan kurtosis pada sebaran normal adalah 0. Sebaran dengan ekor yang lebih panjang ke kanan ialah sebaran yang menjulur positif, sedangkan sebaran dengan ekor yang lebih panjang ke kiri ialah sebaran yang menjulur negatif. Sebaran yang simetris seperti sebaran normal mempunyai nilai kemenjuluran sama dengan 0. Sebaran dengan ekor yang lebih panjang dari normal (lebih rawan memiliki pencilan) memiliki kurtosis positif, dan mereka dengan ekor lebih pendek dari normal (cenderung memiliki sedikit pencilan) memiliki kurtosis negatif.

Kemenjuluran mempengaruhi selang kepercayaan. Kemenjuluran dan kurtosis pada data dapat mempengaruhi selang kepercayaan, akan tetapi kemenjuluran mempunyai efek yang lebih besar dibanding kurtosis. Ketika suatu data menyebar normal, selang kepercayaan dua arah dari sebaran t - berada pada selang/cakupan yang benar, galat yang sama besar pada data yang tinggi dan rendah. Ketika data dari sebaran mempunyai nilai kemenjuluran sama dengan nol, maka selang kepercayaan pada uji dua arah juga masih berada di sekitar nilai/cakupan yang benar, tetapi galat cenderung ke satu sisi atau sisi yang lainnya, tidak sama besar secara merata pada data yang besar dan kecil. Selang kepercayaan pada uji satu arah untuk rata-rata bisa menjadi masalah yang serius.

2. Pencilan, kekekaran, dan resistensi

Suatu pencilan dapat mempengaruhi rata-rata perlakuan juga KTG. Sehingga inferensi yang diperoleh jika memasukkan dan mengeluarkan pencilan akan berbeda. Hal ini mungkin berguna untuk membedakan untuk membedakan antara kekekaran (validitas) dan resistensi (untuk pencilan). Kekekaran pada validitas artinya bahwa prosedur tersebut memberikan kesimpulan yang masih mendekati benar, bahkan ketika beberapa asumsi kita (seperti normalitas) tidak terpenuhi. Jadi dapat dikatakan bahwa uji-F pada ANOVA masih kekar, karena nilai uji-F 5% menolak nol di sekitar 5% dari seluruh contoh ketika hipotesis nol benar, bahkan ketika data yang diperoleh agak tidak normal. Suatu prosedur dikatakan resisten jika satu atau beberapa nilai data individu tidak mengubah inferensi dari suatu data. Walaupun uji F kekar pada validitasnya, tetapi uji-F tidak resisten terhadap pencilan.

4.2 Kehomogenan ragam

Terkadang, sisaan tidak konstan di nilai \hat{y} yang berbeda (dalam kata lain, di taraf faktor berbeda-beda). Jika penurunan ANOVA yang diberikan di Montgomery (2017) diteliti, penurunan nilai harapan KT_G sebagai penduga σ^2 memerlukan asumsi kehomogenan ragam. Oleh karena itu, jika asumsi tersebut tidak terpenuhi, penduga ragam yang dimiliki bukan penduga terbaik. Ini berarti penduga efek perlakuan, kelompok, dan lainnya belum tentu efisien (ragam belum tentu terkecil) dan hasil uji-F dapat salah. Namun, penduga tersebut masih tak bias karena ketidakbiasan muncul dari nilai harapan e_{ij} , dan sifat tersebut dipastikan MKT.

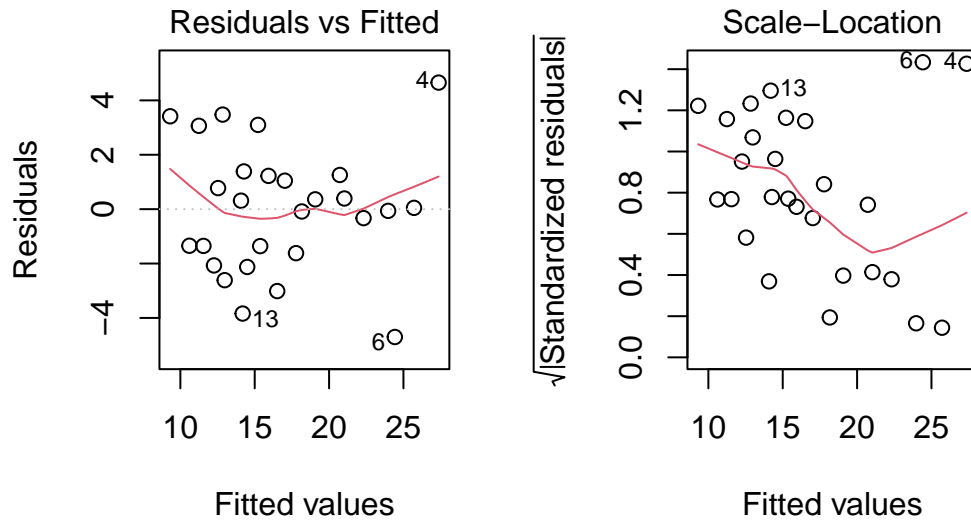
Dalam rancangan percobaan dengan jumlah ulangan sama di semua perlakuan, uji-F tidak terlalu dipengaruhi ragam tak homogen. Namun, jika jumlah ulangan tidak sama atau jika ada ragam yang jauh lebih besar dari ragam lainnya, masalah tersebut lebih serius. Jika ragam besar muncul di perlakuan dengan ulangan sedikit, α aktual lebih besar dari yang diinginkan (sehingga lebih mungkin menolak H_0 jika H_0 benar). Jika ragam besar muncul di perlakuan dengan ulangan banyak α aktual lebih kecil dari yang diinginkan (peneliti tanpa sengaja memilih uji yang terlalu ketat). Jika model acak digunakan, inferensia mengenai ragam akan terganggu walaupun ulangan sama digunakan.

Blanca *et al.* (2017) mendiskusikan kekekaran uji-F di berbagai situasi. Sama seperti Montgomery, ragam tak sama tidak mengganggu uji-F saat ulangan sama. Namun, uji-F terganggu jika jumlah ulangan tak sama saat rasio antara ragam perlakuan terkecil dan terbesar lebih dari 1.5.

4.2.1 Prosedur Eksploratif

Prosedur eksploratif yang dapat dilakukan adalah plot residual vs. fitted dan plot scale-location. Plot ini dapat dilihat dengan menggunakan fungsi plot(hasilmodel). Fungsi tersebut sebenarnya mengeluarkan berbagai jenis plot, seperti plot kuantil-kuantil dari residual (yang akan berguna saat melihat normalitas dari residual). Implementasi plotting tersebut di R adalah:

```
par(mfrow=c(1,2)) #layout: mfrow= 1 baris, 2 kolom
plot(aovRAKL, which=c(1,3)) #plot 1 (residual-fitted) dan 3 (scale-location)
```

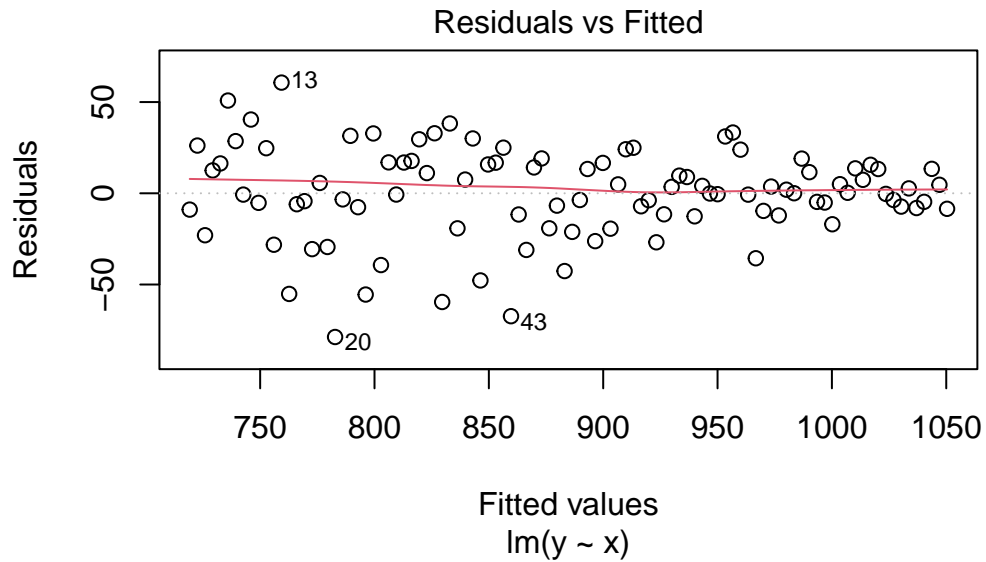


Perlu dilihat dua hal dari plot residual vs fitted. Pertama, garis merah di plot tersebut seharusnya lurus di nol. Kedua, titik-titik di plot tersebut menyebar seragam, tidak membentuk pola tertentu. Terlihat bahwa garis merah melengkung di kedua ujung plot residual vs. fitted. Selain itu, titik-titik tampak membentuk pola corong terbalik. Hal tersebut dapat terlihat dari data yang sengaja dibuat memiliki ragam tak homogen. Demonstrasi ini diadaptasi dari [Hanck *et al.* \(2016\)](#).

```
set.seed(101)

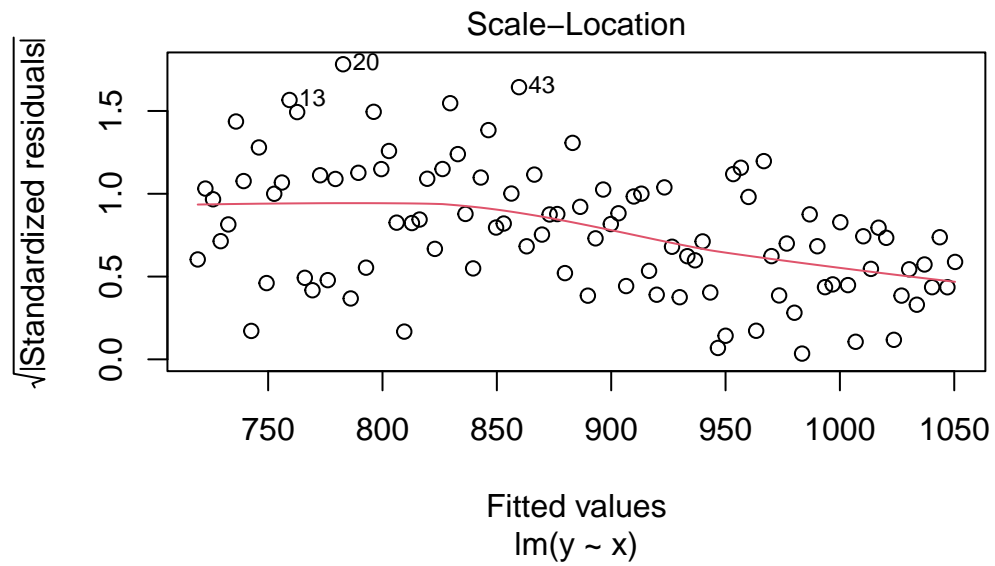
x <- rep(seq(1,100)) #vektor x dari 1-100
e <- c()
e[1:25] <- rnorm(25, sd = 40)
e[26:50] <- rnorm(25, sd = 30)
e[51:75] <- rnorm(25, sd = 20)
e[76:100] <- rnorm(25, sd = 10)
# vektor galat, saat x kecil, ragam galat dibuat besar
y <- 720 + 3.3 * x + e
mod <- lm(y ~ x)

plot(mod, which=1) #plot model
```



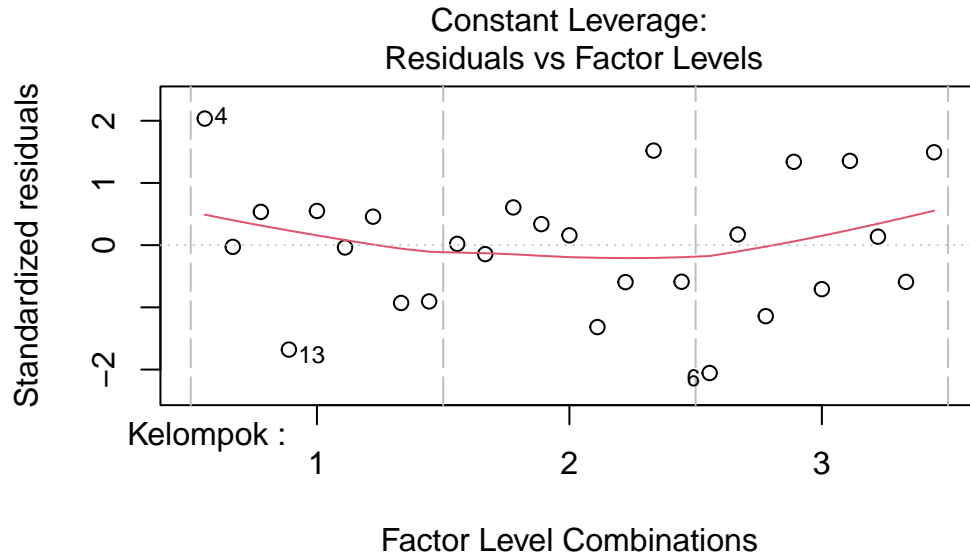
Selain itu, heteroskedastisitas juga terlihat di plot scale-location. Seharusnya garis merah di plot scale-location lurus dan titik-titik terdistribusi homogen. Walaupun ada dua amatan yang tak sesuai dengan pola tersebut, terlihat bahwa ada pola menurun di plot tersebut saat x membesar. Jika kembali ke contoh data dengan ragam tak sama sebelumnya:

```
plot(mod, which=3)
```



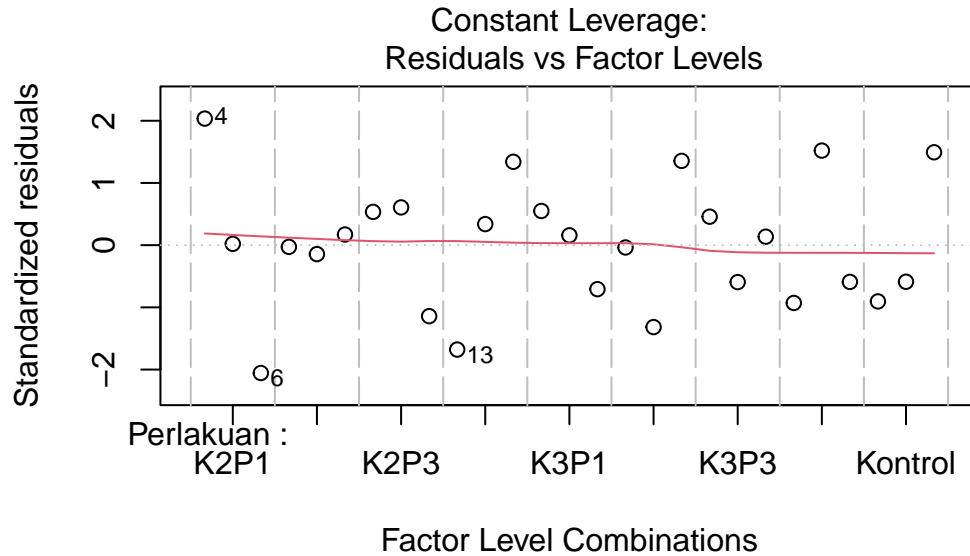
Pola menurun tersebut juga terlihat. Plot terakhir yang dapat digunakan untuk mendiagnosis ragam tak seragam adalah plot residual vs level faktor:

```
plot(aovRAKL, which=5) #plot ke-5
```

Plot ini menunjukkan sebaran residual per kelompok, sedangkan sebaran residual per perlakuan dapat dilihat di:

```
aovRAKL2<-aov(Produksi~Perlakuan+Kelompok,data=DataRAKL) #ubah urutan anova. peubah pertama
plot(aovRAKL2,which=5)
```



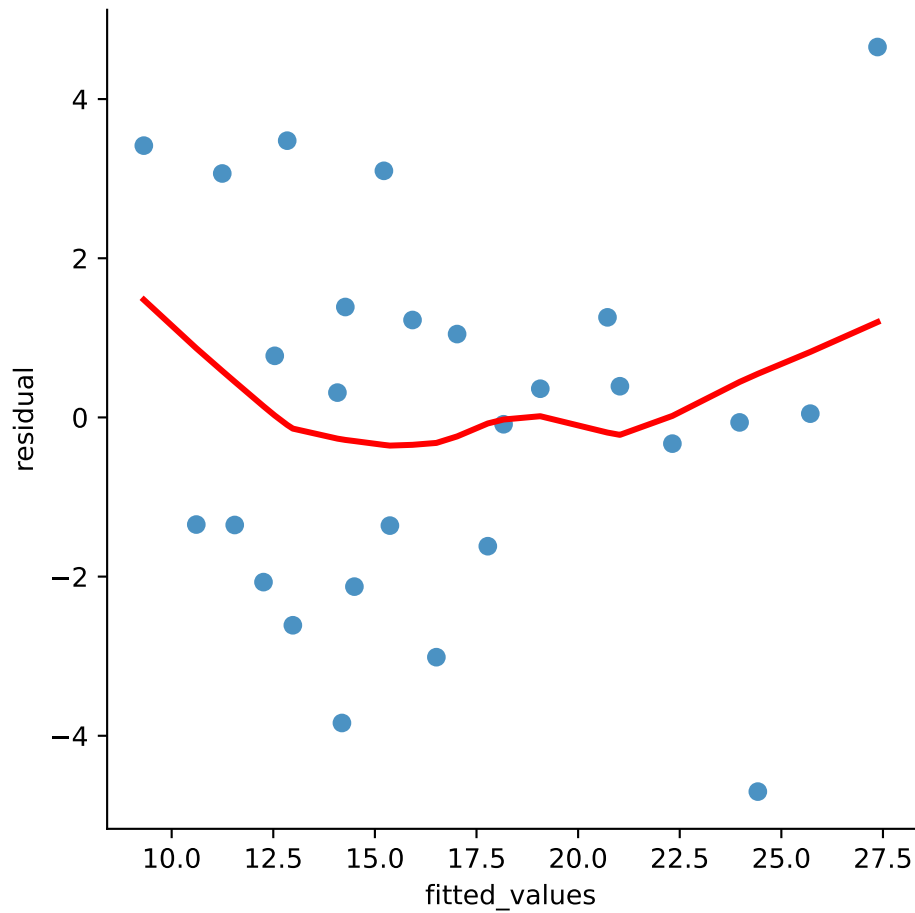
Ketidakhomogenan ragam tidak terlihat di kedua plot ini.

Buat plot residual vs fitted terlebih dahulu. Pertama, buat data frame dengan nilai-nilai yang diinginkan:

```
temp_data = pd.DataFrame(dict(fitted_values=RAKL_lm.fittedvalues, residual=RAKL_lm.resid))
#ambil fitted values dan residual dengan hasilANOVA.fittedvalues/resid
```

Buat plot dari data tersebut. Buat argumen lowess=True agar garis merah terlihat seperti di R:

```
graph = sns.lmplot(x='fitted_values', y='residual', data=temp_data, lowess=True, line_kws=
plt.show())
```

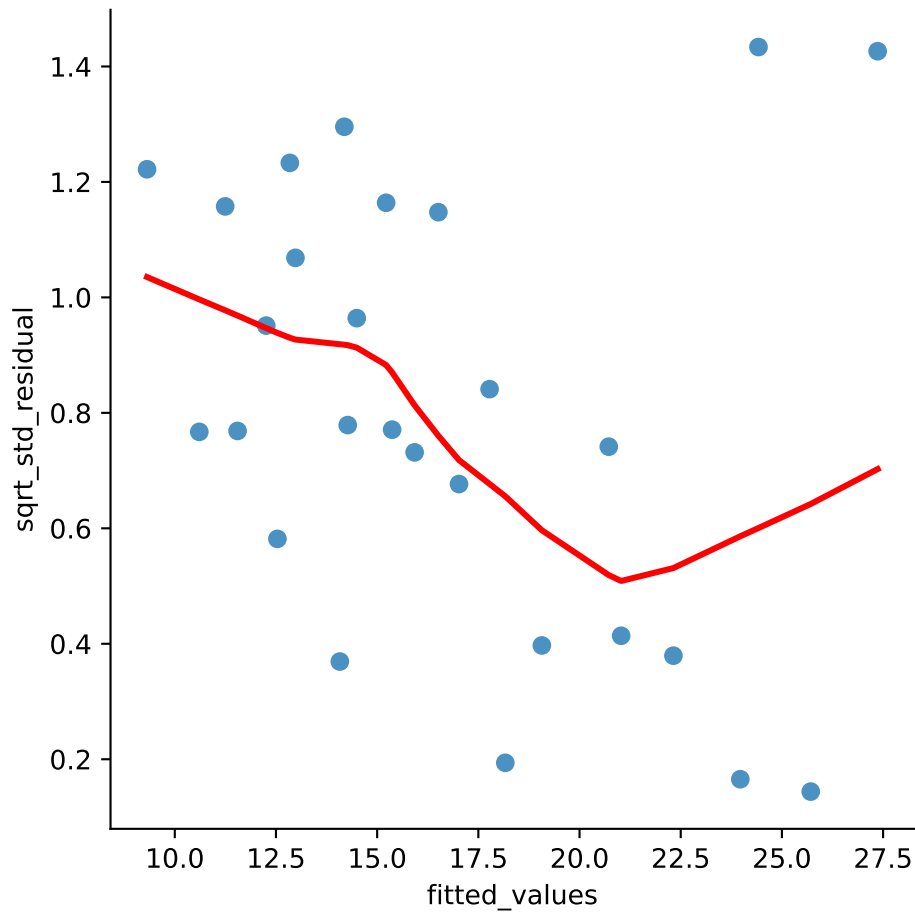


Pola corong terbalik juga terlihat di plot ini. Lalu, untuk membuat plot scale-location di python, ambil *studentized residual* dari model tersebut dulu:

```
sqrt_std_residual = np.sqrt(np.abs(RAKL_lm.get_influence().resid_studentized_internal))
#hasil_model.get_influence().resid_studentized_interal: ambil studentized residual
#np.sqrt: akar
temp_data_scaleloc = pd.DataFrame(dict(fitted_values=RAKL_lm.fittedvalues,
                                       sqrt_std_residual=sqrt_std_residual)) #masukkan dataframe
```

Lalu, plot-kan:

```
graph = sns.lmplot(x='fitted_values', y='sqrt_std_residual', data=temp_data_scaleloc, lowess=True)
plt.show()
```



Pola menurun terlihat juga di plot ini.

4.2.2 Uji formal untuk kehomogenan ragam

Khalayak yang berbeda dapat memberikan tafsiran berbeda untuk tiap plot, khususnya jika ketidakhomogenan ragam tidak seekstrim di contoh sebelumnya. Pola-pola tertentu belum tentu terlihat, khususnya jika audiens kurang teliti. Oleh karena itu, perlu uji formal untuk kehomogenan ragam. Semua uji yang akan diperlihatkan memiliki hipotesis:

$$H_0 : \sigma_1^2 = \sigma_2^2 = \dots = \sigma_t^2 = \sigma^2 \text{ (ragam homogen)}$$

$$H_1 : \text{Setidaknya satu } \sigma_i^2 \neq \sigma^2 \text{ (ragam tak homogen)}$$

4.2.2.1 Uji yang didasari asumsi normalitas: Bartlett dan Hartley

Uji Bartlett memiliki statistik uji:

$$\chi_0^2 = 2.3026 \frac{q}{c}$$
$$q = (N - t) \log_{10} S_p^2 - \sum_{i=1}^t (r_i - 1) \log_{10} S_i^2$$
$$c = 1 + \frac{1}{3(t-1)} \left[\sum_{i=1}^t (n_i - 1)^{-1} - (N - a)^{-1} \right]$$
$$S_p^2 = \frac{\sum_{i=1}^t (r_i - 1) S_i^2}{N - t}$$

Dengan S_i^2 adalah ragam sampel perlakuan atau kelompok ke-i, N jumlah total unit percobaan. Arti dari rumus tersebut adalah upaya membandingkan penduga ragam jika homogen S_p^2 dengan jumlah ragam sebenarnya. Jika ragam homogen, statistik q akan mendekati nol. Jika heterogen, statistik q akan besar sehingga H_0 akan ditolak jika $\chi_0^2 > \chi_{\alpha; t-1}^2$

Sedangkan uji Hartley memiliki statistik uji:

$$F_{max} = \frac{\max_i [S_i^2]}{\min_i [S_i^2]}$$

Atau rasio antara ragam maksimum dan ragam minimum. Uji ini mengharuskan jumlah ulangan sama. Penduga ragam menyebar chi-kuadrat jika peubah acak tersebut menyebar normal¹. Jika jumlah ulangan sama di suatu perlakuan atau kelompok, maka derajat bebas tiap penduga ragam sama. Pembagian dua peubah chi-kuadrat dengan derajat bebasnya menghasilkan peubah yang menyebar F. Karena jumlah ulangan sama, angka tersebut saling terbagi.

Jika ada beberapa perlakuan yang digunakan, atau beberapa kelompok, dapat dilakukan uji untuk tiap perlakuan tersebut. Implementasi uji tersebut di R sebagai berikut:

```
bartlett.test(Produksi ~ Perlakuan, data = DataRAKL) #uji kehomogenan ragam di tiap perlakuan
```

```
Bartlett test of homogeneity of variances
```

```
data: Produksi by Perlakuan  
Bartlett's K-squared = 5.7463, df = 8, p-value = 0.6756
```

¹Dapat dilihat di buku Teori Statistika seperti Ramachandran dan Tsokos

```
bartlett.test(Produksi ~ Kelompok, data = DataRAKL) #uji kehomogenan ragam di tiap kelompok
```

Bartlett test of homogeneity of variances

data: Produksi by Kelompok

Bartlett's K-squared = 2.5611, df = 2, p-value = 0.2779

Sedangkan uji Hartley:

```
library(PMCMRplus)
hartleyTest(Produksi~Perlakuan,data=DataRAKL)
```

Hartley's maximum F-ratio test of homogeneity of variances

data: Produksi by Perlakuan

F Max = 22.226, df = 2, k = 9, p-value = 0.6461

```
hartleyTest(Produksi~Kelompok,data=DataRAKL)
```

Hartley's maximum F-ratio test of homogeneity of variances

data: Produksi by Kelompok

F Max = 3.1755, df = 8, k = 3, p-value = 0.2647

Kedua uji tersebut tidak menolak H_0 di $\alpha = 5\%$. Oleh karena itu, kesimpulan dari dua uji tersebut adalah bahwa ragam homogen. Implementasi uji Bartlett di python menggunakan scipy sebagai berikut:

```
from scipy.stats import bartlett
kel1 = RAKLpy.query('Kelompok == 1')['Produksi'] #ambil nilai peubah respon di tiap kelompok
kel2 = RAKLpy.query('Kelompok == 2')['Produksi'] #sintaks: dataframe.query('kondisi')['peubah']
kel3 = RAKLpy.query('Kelompok == 3')['Produksi']

stat, p = bartlett(kel1, kel2, kel3) #lakukan uji
```

```
print("Statistik uji:", stat, "| P-value:", p) #print hasil uji
```

Statistik uji: 2.5610747969217615 | P-value: 0.27788792377685606

Implementasi ini cukup membosankan karena harus mengambil data tiap kelompok secara manual. Implementasi menggunakan pinguin tidak memerlukan hal tersebut. Implementasi ini lebih baik jika ada banyak kelompok. Misal, ada tujuh perlakuan berbeda. Pengambilan data untuk tiap perlakuan akan membosankan. Karena itu, gunakan pinguin:

```
import pinguin as pg

pg.homoscedasticity(RAKLpy, dv='Produksi',
                    group='Perlakuan',
                    method='bartlett') #(data, peubah respon (y), faktor, metode)
```

	T	pval	equal_var
bartlett	5.746278	0.675628	True

Kesimpulan uji menggunakan R ataupun python sama. Ragam homogen.

4.2.3 Uji Levene dan Brown-Forsythe

Uji Bartlett dan Hartley bergantung pada asumsi normalitas. Jika sebaran galat tidak normal, misal menyebar Student-t atau chi-kuadrat, uji tersebut akan cenderung mengambil keputusan yang salah. Uji tersebut dapat secara empiris memiliki α yang lebih tinggi dari yang diinginkan dan memiliki kuasa uji lebih rendah dari alternatif yang tidak mengasumsikan normalitas.

Alternatif uji yang tidak mengasumsikan normalitas adalah uji Levene, dengan statistik uji:

$$W = \frac{(N-t)}{(t-1)} \frac{\sum_{i=1}^t r_i (\bar{Z}_{i.} - \bar{Z}_{..})^2}{\sum_{i=1}^t \sum_{j=1}^r (Z_{ij} - \bar{Z}_{i.})^2}$$

Uji Levene menghasilkan sekeluarga uji dengan statistik uji yang mirip, hanya dibedakan oleh pilihan Z_{ij} . Uji Levene menggunakan rata-rata perlakuan ke- i $\bar{Y}_{i.}$, sedangkan ada juga uji yang menggunakan rata-rata terpankaskan 10% $\hat{Y}_{i.}$. Uji Brown-Forsythe menggunakan median $\tilde{Y}_{i.}$. Jika disimpulkan:

$$\begin{aligned}\text{Levene} &: Z_{ij} = |y_{ij} - \bar{y}_{i.}| && (\text{Mean}) \\ \text{Levene (trimmed)} &: Z_{ij} = |y_{ij} - \hat{y}_{i.}| && (\text{Mean terpangkas}) \\ \text{Brown-Forsythe} &: Z_{ij} = |y_{ij} - \bar{y}'_{i.}| && (\text{Median})\end{aligned}$$

Brown dan Forsythe (1974) membandingkan kuasa uji dan α empiris beberapa uji homogenitas ragam di berbagai situasi dengan simulasi. Keluarga uji Levene memiliki kuasa uji lebih rendah jika data berdistribusi normal, tetapi di distribusi Student-t dan chi-kuadrat kuasa uji Levene lebih tinggi. Uji Levene tidak kekar pada data dengan penjumlahan ekstrim karena memiliki tingkat kesalahan jenis I (tolak H_0 jika H_0 benar) lebih tinggi secara empiris daripada yang dispesifikasi peneliti. Uji Brown-Forsythe sangat konservatif, bahkan dengan tingkat kesalahan jenis I (α) empiris lebih rendah dari yang dispesifikasi. Uji Levene dengan mean terpangkas memiliki α empiris paling dekat dengan yang diinginkan. Penemuan ini direplikasi di penelitian lainnya dan disimpulkan di Conover *et al.* (2018).

Oleh karena itu, dalam menentukan uji paling tepat untuk kehomogenan ragam, perlu diteliti sebaran dari sisaan. Jika sebaran normal, uji Bartlett ataupun Hartley paling baik. Jika sebaran tidak normal tetapi tidak menjulur ekstrim, uji Levene memiliki kuasa uji paling tinggi daripada uji lainnya. Jika sebaran menjulur ekstrim, uji Levene dengan mean terpangkas atau uji Brown-Forsythe paling kekar.

Implementasi uji tersebut di R adalah sebagai berikut:

```
library(car)
```

Loading required package: carData

```
leveneTest(Produksi~Perlakuan, data=DataRAKL, center = median) #default (Brown Forsythe)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  8  0.5855 0.7771
      18
```

```
leveneTest(Produksi~Kelompok, data=DataRAKL, center = mean) # Levene (mean)
```

```
Levene's Test for Homogeneity of Variance (center = mean)
      Df F value Pr(>F)
group  2  0.8113 0.4561
      24
```



```
leveneTest(Produksi~Perlakuan, data=DataRAKL, center = mean,trim=0.1) # Levene (trimmed)
```

Levene's Test for Homogeneity of Variance (center = mean: 0.1)

```
      Df F value Pr(>F)
group  8  1.0536 0.4351
      18
```

Karena p-value lebih dari $\alpha = 5\%$, ragam dapat dianggap homogen untuk kelompok dan perlakuan di taraf tersebut. Sedangkan implementasi uji tersebut di Python:

```
from scipy.stats import levene
statlev, plev = levene(kel1, kel2, kel3,center="median") #center median
statlev_trim, plev_trim = levene(kel1, kel2, kel3,center="mean", proportiontocut=0.1) #cen

print("Statistik uji (Brown-Forsythe):", statlev, "| P-value:", plev) #print hasil uji
```

Statistik uji (Brown-Forsythe): 0.7589478077456477 | P-value: 0.47906740355022637

```
print("Statistik uji (Trim):", statlev_trim, "| P-value:", plev_trim) #print hasil uji
```

Statistik uji (Trim): 0.8113082087822503 | P-value: 0.45609278276162774

Kode python untuk uji ini lebih pendek karena vektor hasil per kelompok sudah diambil sebelumnya untuk uji Bartlett. Jika menggunakan pingouin:

```
import pingouin as pg
pg.homoscedasticity(RAKLpy, dv='Produksi',
                    group='Perlakuan',
                    method='levene') #(data, peubah respon (y), faktor, metode)
```

```
      W      pval  equal_var
levene 0.585459 0.777083      True
```

Sayangnya, hanya dapat dilakukan uji Brown-Forsythe (atau, uji Levene menggunakan median). Walaupun implementasi pingouin lebih sederhana, metode ini kurang fleksibel.

4.2.4 Plot vs. uji

Terlihat bahwa hasil uji formal menyimpulkan bahwa ragam sama, sedangkan plot residual vs. fitted dan scale-location menunjukkan pola heteroskedastis. Mungkin, amatan 6 dan 14 berpengaruh. Coba hilangkan amatan tersebut:

```
DataRAKL_tanpaoutlier<-DataRAKL[c(-6,-14),]  
bartlett.test(Produksi ~ Perlakuan, data = DataRAKL_tanpaoutlier) #uji kehomogenan ragam di t
```

Bartlett test of homogeneity of variances

data: Produksi by Perlakuan
Bartlett's K-squared = 2.5855, df = 8, p-value = 0.9576

```
bartlett.test(Produksi ~ Kelompok, data = DataRAKL_tanpaoutlier) #uji kehomogenan ragam di t
```

Bartlett test of homogeneity of variances

data: Produksi by Kelompok
Bartlett's K-squared = 2.4855, df = 2, p-value = 0.2886

```
leveneTest(Produksi ~ Perlakuan, data = DataRAKL_tanpaoutlier) #uji kehomogenan ragam di t
```

Levene's Test for Homogeneity of Variance (center = median)

	Df	F value	Pr(>F)
group	8	0.3319	0.941
	16		

```
leveneTest(Produksi ~ Kelompok, data = DataRAKL_tanpaoutlier) #uji kehomogenan ragam di ti
```

Levene's Test for Homogeneity of Variance (center = median)

	Df	F value	Pr(>F)
group	2	0.8511	0.4405
	22		

Uji formal masih menandakan ragam sama. Lalu, bagaimana cara hasil plot residual vs fitted dijelaskan?

Pertama, plot residual per kelompok dan perlakuan relatif sama. Karena uji hanya dapat dilakukan kepada kelompok atau perlakuan, bukan kombinasinya, hasil ini sesuai dengan plot tersebut. Kedua, pola tersebut mungkin muncul karena observasi sedikit di fitted values yang besar sehingga terlihat berpola padahal tidak. Ketiga, pola tersebut mungkin muncul karena interaksi antara kelompok dan perlakuan yang menurut uji Tukey ada. Oleh karena itu, uji yang terpisah di perlakuan atau kelompok tidak dapat mendeteksi hal tersebut, tapi uji yang melihat kombinasi residual dapat mendeteksi hal tersebut. Mungkin dapat dibandingkan dengan uji Bresuch-Pagan (yang menjadi uji standar heteroskedastisitas di regresi):

```
library(lmtest)
```

```
Loading required package: zoo
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':
```

```
as.Date, as.Date.numeric
```

```
bptest(Produksi~Perlakuan+Kelompok,data=DataRAKL)
```

```
studentized Breusch-Pagan test
```

```
data:  Produksi ~ Perlakuan + Kelompok
```

```
BP = 13.784, df = 10, p-value = 0.1831
```

Walaupun masih taktolak H_0 , hasil uji Breusch-Pagan memiliki p-value lebih rendah dari uji lain. Oleh karena itu, aman untuk mengasumsikan ragam homogen di kasus ini. Selain itu karena jumlah ulangan sama, heterogenitas ragam tak berpengaruh besar, sesuai diskusi sebelumnya.

4.3 Galat percobaan saling bebas

Kebebasan galat merupakan asumsi yang paling penting dari ANOVA dan juga yang paling sulit diatasi ketika asumsi tersebut tidak terpenuhi. Asumsi ragam galat konstan merupakan asumsi yang terpenting kedua untuk dipenuhi setelah asumsi kebebasan galat. Akibat dari ragam galat yang tidak konstan akan mempengaruhi kesimpulan yang kita peroleh. Namun, ragam galat yang tidak konstan dapat diatasi pada banyak situasi, misal menggunakan transformasi.

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}$$

Pada model linier yang kita gunakan pada persamaan di atas, asumsi kenormalan menjadi tidak begitu penting untuk dipenuhi, terutama jika ukuran contoh yang diperoleh besar dan Teorema Limit Pusat dapat dipakai. Namun terdapat model-model tertentu yang sangat bergantung pada normalitas (Nur Hilal A. Syahrir, Statistika Terapan IPB).

Autokorelasi muncul ketika hasil pada amatan yang berdekatan cenderung terlalu mirip (korelasi positif) atau terlalu berbeda (korelasi negatif). Autokorelasi seringkali menjadi penyebab ketidakbebasan pada galat. Alasan amatan tidak saling bebas adalah:

1. Pengukuran berulang dilakukan ke subjek yang sama
2. Amatan berkorelasi dalam waktu
3. Amatan berkorelasi dalam tempat
4. Pengaturan sistematis dari unit percobaan

Contoh pengukuran berulang dan amatan yang berkorelasi dalam waktu adalah jika tiga pohon dengan varietas berbeda diamati tingginya. Misal, tiga pohon tersebut diamati tingginya setelah satu, dua, dan tiga tahun. Jika amatan di varietas ke- i di tahun ke- j menjadi baris di data, amatan-amatan tersebut akan tidak bebas, melainkan berkorelasi dengan waktu. Oleh karena itu, amatan tiap tahun harus dibuat kolom sendiri agar tiap baris (pohon) saling bebas. Namun, jika hanya diamati satu pohon untuk tiap varietas, tidak akan ada derajat bebas bagi ANOVA. Kasus ini menunjukkan salah satu kendala yang muncul saat penanganan amatan tak bebas.

Jika peneliti tersebut menyadari dari awal bahwa amatan suatu pohon di tahun berbeda tidak saling bebas, seharusnya peneliti menyiapkan beberapa pohon dari satu varietas. Jika ada pengaturan tersebut, derajat bebas cukup untuk melakukan ANOVA.

Contoh dari amatan yang berkorelasi dalam tempat adalah perlakuan A yang ditanam di lereng suatu bukit dan perlakuan lainnya ditanam di puncak bukit. Oleh karena berada di tempat geografis yang sama, amatan perlakuan A tidak saling bebas satu sama lain.

Pengaturan sistematis muncul ketika pengacakan tidak dilakukan dengan benar. Akan muncul pola dari urutan unit percobaan.

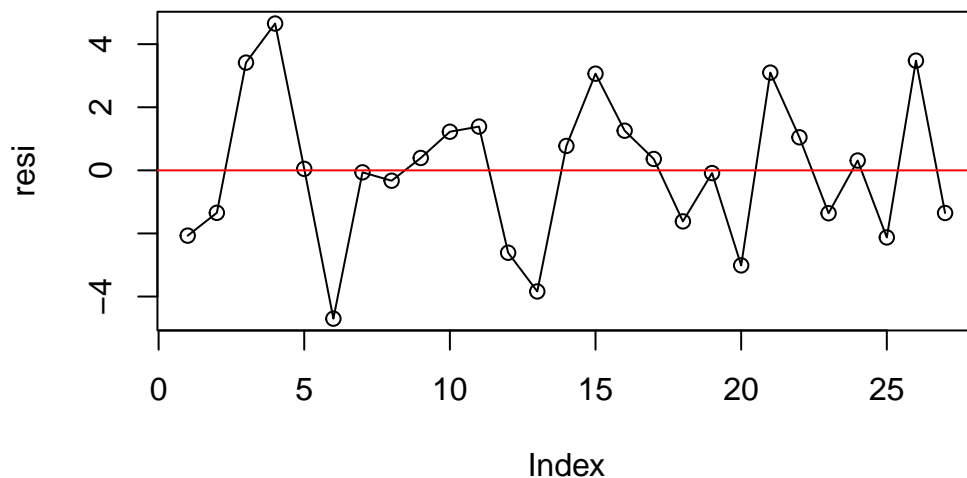
Sebelum melakukan eksplorasi ataupun uji, pastikan **urutan dari amatan masuk akal**. Dalam studi kasus ini, dianggap urutan amatan dari DataRAKL sesuai dengan suatu urutan waktu. Misal, baris 1 diamati sebelum baris 2, dan seterusnya. Jika **urutan tidak memiliki arti**, maka data dapat dengan mudah diubah urutannya agar tidak berkorelasi satu sama lain, seperti di [contoh ini] (<https://stats.stackexchange.com/questions/326305/ran-the-durbinwatson-test-in-r-and-got-p-value-0-for-non-time-series-data-is>).

4.3.1 Prosedur Eksploratif

4.3.1.1 Plot residual vs order

Memeriksa adanya autokorelasi secara eksploratif dilakukan dengan melihat plot antara sisaan dengan urutan waktu (*plot of residuals in time sequence*). Akan diteliti tren dari plot tersebut - menyebar secara acak atau tidak berpola apapun. Dapat juga melihat pola *plot of residual versus order*, khususnya apakah plot tersebut stasioner. Implementasi plotting tersebut di R adalah:

```
resi <- residuals(object = aovRAKL) #ambil residual
# Plot Sisaan vs Order
plot(resi, type = 'o'); abline(h = 0, col = 'red')
```

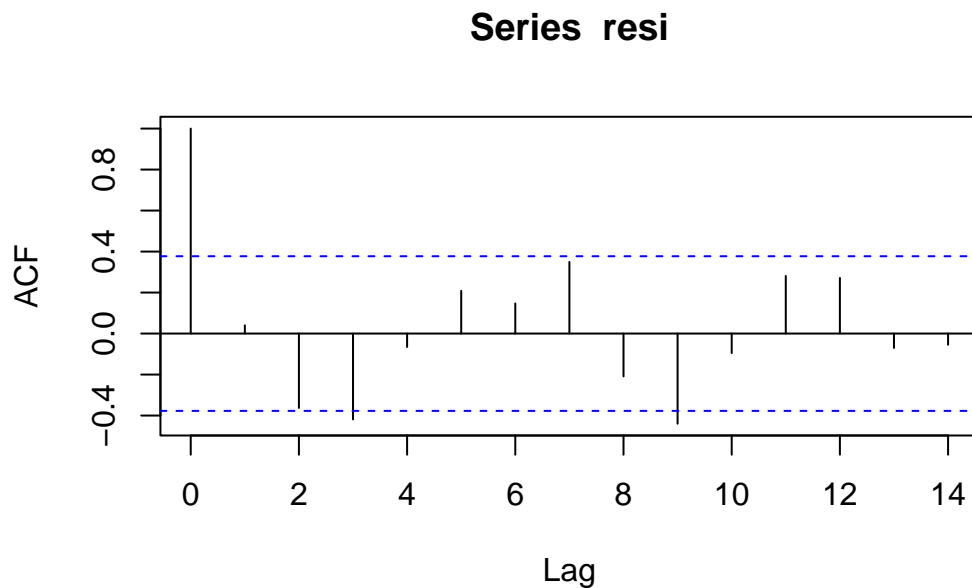


Dari gambar plot sisaan vs order (plot of residual versus order) terlihat membentuk trend yang datar atau hampir datar sehingga asumsi galat percobaan saling bebas terpenuhi. Namun, titik tersebar membangun pola melebar atau menyempit (bentuk terompet) sehingga data mempunyai pola yang tidak stasioner.

4.3.1.2 ACF dan PACF

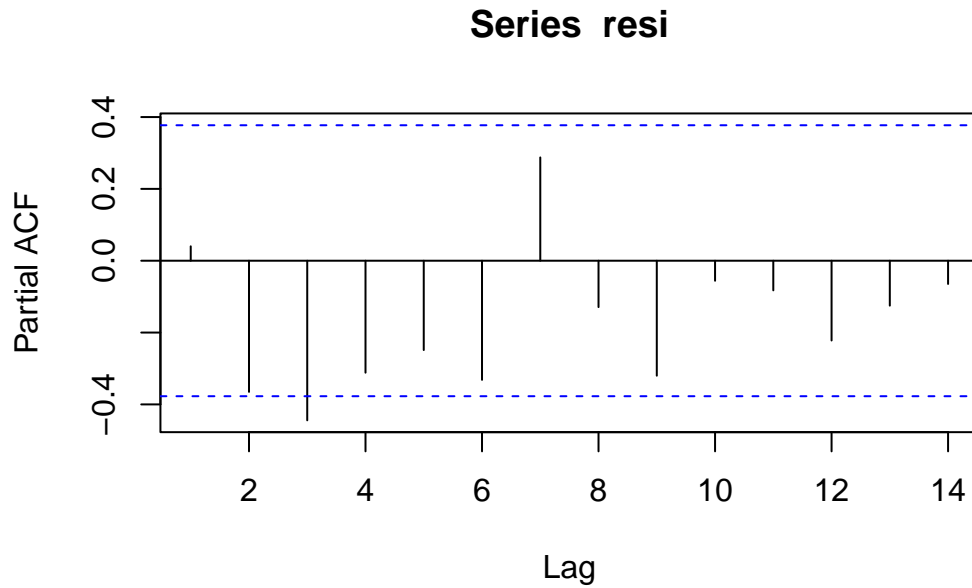
Selain plot tersebut, dapat dilihat plot autokorelasi (ACF) dan plot autokorelasi parsial (PACF). Lag adalah beda urutan suatu sisaan dengan sisaan sebelumnya. Misal, lag 2 berarti tiap sisaan dibandingkan dengan sisaan dua observasi sebelumnya. Jika garis vertikal di tiap lag lebih tinggi dari garis biru horizontal, maka dianggap terjadi autokorelasi di lag tersebut.

```
acf(resi)
```



Plot korelasi parsial mengeluarkan pengaruh dari lag-lag lain. Aturan keputusan dari plot tersebut sama dengan plot ACF:

```
pacf(resi)
```



Lag nol di plot ACF dapat diabaikan karena korelasi antara observasi dengan dirinya sendiri pasti satu. Namun, terlihat bahwa di beberapa lag bukan nol garis vertikal melewati garis horizontal sehingga dapat disimpulkan bahwa galat tidak bebas.

4.3.2 Uji formal untuk galat percobaan saling bebas

Uji formal untuk menentukan kebebasan galat percobaan menggunakan Runs Test (Uji Run-tunan) dan Uji Durbin Watson.

4.3.2.1 Uji Durbin-Watson

Uji formal yang sederhana yang biasa digunakan untuk mengecek autokorelasi adalah Uji Durbin Watson. Uji Durbin Watson hanya dapat menguji ketidakbebasan galat percobaan pada lag 1. Dalam uji tersebut, anggap bahwa sisaan ke- t adalah $e_t = \rho e_{t-1} + v_t$, di mana v_t merupakan komponen sisaan yang memenuhi asumsi kebebasan dan kehomogenan. Oleh karena itu, hipotesis yang diuji adalah:

$$H_0 : \rho = 0$$

$$H_1 : \rho \neq 0$$

Jika e_k adalah residual yang diurut berdasarkan urutan waktu, statistik Durbin-Watson adalah:

$$DW = \frac{\sum_{k=2}^n (e_k - e_{k-1})^2}{\sum_{k=1}^n e_k^2}$$

Interpretasi yang tepat dari nilai statistik uji Durbin-Watson, jika dilakukan secara manual, cukup sulit. Sebaran tepat statistik uji Durbin Watson bergantung pada matriks peubah bebas X sehingga tidak ada satu tabel yang mengandung nilai kritis statistik Durbin Watson. Jika tidak ada autokorelasi, DW berada di sekitar 2. Autokorelasi positif akan membuat DW kurang dari 2, dan jika terdapat autokorelasi negatif maka nilai DW akan lebih besar dari 2.

Sebagai perhitungan kasar, autokorelasi terjadi ketika DW berada di luar rentang 1,5 sampai 2,5. Nilai tersebut cukup besar dalam melihat efek yang nyata pada teknik inferensi. Perhatikan bahwa DW sendiri acak dan mungkin di luar jangkauan 1,5 sampai 2,5 bahkan jika residual tidak berkorelasi. Untuk kumpulan data dengan amatan yang panjang dari perlakuan yang sama, ragam dari DW kurang dari $4/N$.

Namun, implementasi uji Durbin-Watson di berbagai software statistika, termasuk R, dapat diinterpretasikan dengan mudah karena software tersebut dapat menghitung sebaran statistik uji Durbin Watson secara eksak. Matriks peubah bebas X dapat menjadi input software dan p-value eksak dapat dikeluarkan oleh software tersebut. Uji Durbin-Watson dapat diimplementasikan melalui package `lmtest` atau `car`.

```
library(lmtest)
library(car)
lmtest::dwtest(aovRAKL,alternative="two.sided", iter=15, tol=1e-10) #H1: rho tak sama deng
```

Durbin-Watson test

```
data: aovRAKL
DW = 1.8764, p-value = 0.04881
alternative hypothesis: true autocorrelation is not 0
```

```
#iter=iterasi algoritma pan, tol=toleransi
dwtest(aovRAKL,alternative="greater") #H1: rho lebih dari nol
```

Durbin-Watson test


```
data: aovRAKL
DW = 1.8764, p-value = 0.02441
alternative hypothesis: true autocorrelation is greater than 0
```

```
dwtest(aovRAKL,alternative="less") #H1: rho kurang dari nol
```

Durbin-Watson test

```
data: aovRAKL
DW = 1.8764, p-value = 0.9756
alternative hypothesis: true autocorrelation is less than 0
```

```
car::durbinWatsonTest(aovRAKL)
```

```
lag Autocorrelation D-W Statistic p-value
1      0.04019671      1.876365    0.034
Alternative hypothesis: rho != 0
```

Lebih direkomendasikan menggunakan package `lmtest` karena p-value package `lmtest` lebih stabil. Ini terjadi karena algoritma “pan” yang digunakan package tersebut untuk mengatur p-value. Pada dasarnya, algoritma tersebut memastikan p-value konvergen (menuju ke) suatu nilai sehingga tidak berubah-ubah. Seperti dilihat di kode, toleransi telah di-set sehingga perubahan p-value dari satu iterasi ke iterasi selanjutnya lebih sedikit dari toleransi yang diinginkan. Dapat disimulasikan sebagai berikut:

```
carRes<-rep(0,10)
lmtestRes<-rep(0,10)
for (i in seq(1,10)){
  carRes[i]<-car::durbinWatsonTest(aovRAKL)$p
  lmtestRes[i]<-lmtest::dwtest(aovRAKL,alternative="two.sided")$p
}
knitr::kable(data.frame(car=carRes,lmtest=lmtestRes))
```

	car	lmtest
	0.058	0.0488117
	0.052	0.0488117

car	lmtest
0.046	0.0488117
0.036	0.0488117
0.054	0.0488117
0.040	0.0488117
0.054	0.0488117
0.050	0.0488117
0.046	0.0488117
0.038	0.0488117

```
mean(carRes)
```

```
[1] 0.0474
```

Seperti yang dapat dilihat, p-value dari library car sangat tidak stabil, sehingga mungkin saja keputusan uji berubah-ubah. Gunakan lmtest saja. Implementasi di python sebagai berikut:

```
from statsmodels.stats.stattools import durbin_watson

#perform Durbin-Watson test
durbin_watson(RAKL_lm.resid)
```

```
1.8763649943450147
```

Implementasi dari statsmodels tidak dapat memberikan p-value. Oleh karena itu, implementasi dari dwtest, dengan sintaks mirip lmtest dapat menjadi alternatif:

```
from dwtest import dwtest
dw,pval=dwtest(formula='Produksi~Perlakuan+Kelompok',data=RAKLpy,alternative="two_sided",i
print("p-value:",pval,"| DW:",dw)
```

```
p-value: 0.03261757679080751 | DW: 1.8779282884437019
```

Namun, sepertinya implementasi dwtest menghitung nilai statistik uji DW dengan kurang tepat - statistik uji tersebut sedikit berbeda dari keluaran statsmodels atau R. Dari keluaran di atas dapat dilihat bahwa pada uji Durbin Watson ini memiliki nilai p-value $< \alpha = 0.05$ maka Tolak H_0 , artinya cukup bukti untuk mengatakan bahwa ada autokorelasi. Oleh karena

itu, karena autokorelasi seringkali menjadi penyebab ketidakbebasan pada galat maka asumsi galat percobaan saling bebas belum terpenuhi.

Uji Durbin-Watson memiliki beberapa kelemahan:

1. Seperti yang dijelaskan sebelumnya, uji tersebut mengasumsikan autokorelasi pada lag 1 saja.
2. Uji tidak dapat digunakan dalam model autoregresif (memiliki peubah tak bebas dari periode sebelumnya)
3. Uji sensitif terhadap asumsi normalitas.

4.3.2.2 Runs test

Oleh karena itu, dapat digunakan Runs test. Misal ada 15 data, maka jika nilai data naik dari nilai sebelumnya dinotasikan positif (+), dan sebaliknya dinotasikan negatif (-). Jika proses tersebut menghasilkan + + + + - - - + + + - - + + +, ada 5 run, 3 positif dan 2 negatif. Sebuah run merupakan kumpulan tanda positif atau negatif. Jumlah run tersebut menjadi statistik uji runs test, dengan implementasi di R berupa:

```
# Run Test
# H0 (null): The data was produced in a random manner.
# Ha (alternative): The data was not produced in a random manner.
library(snpar)
```

Attaching package: 'snpar'

The following object is masked from 'package:tseries':

```
runs.test
```

```
runs.test(aovRAKL$residuals,exact=T,alternative ="two.sided") #exact=p-value tepat
```

```
Exact runs test
```

```
data: aovRAKL$residuals
Runs = 13, p-value = 0.6951
alternative hypothesis: two.sided
```

Implementasi di python dapat dilihat di [sini](#).

Nilai p dari pengujian ini adalah 0.6951. Karena p-value lebih dari taraf nyata 0.05, maka tak tolak hipotesis nol. Hal ini menunjukkan belum cukup bukti untuk mengatakan bahwa data dihasilkan secara tidak acak.

4.3.2.3 Autokorelasi dengan lag lebih dari 1: Breusch-Godfrey dan modifikasi Durbin-Watson

Seperti dijelaskan sebelumnya, uji Durbin Watson hanya dapat menguji autokorelasi lag 1. Namun, terlihat di plot ACF dan PACF bahwa autokorelasi muncul di lag 2 dan 4. Alternatif sederhana untuk menguji hal tersebut adalah menambah order di fungsi uji Durbin Watson car:

```
durbinWatsonTest(aovRAKL,max.lag=2)
```

```
lag Autocorrelation D-W Statistic p-value
1      0.04019671      1.876365  0.044
2     -0.36296687      2.584265  0.276
Alternative hypothesis: rho[lag] != 0
```

```
durbinWatsonTest(aovRAKL,max.lag=4)
```

```
lag Autocorrelation D-W Statistic p-value
1      0.04019671      1.876365  0.048
2     -0.36296687      2.584265  0.272
3     -0.41913635      2.582042  0.192
4     -0.06558390      1.720913  0.738
Alternative hypothesis: rho[lag] != 0
```

Atau, dapat digunakan uji Breusch-Godfrey. Pada dasarnya, dilakukan regresi residual dengan peubah bebas dan residual lag sebelumnya, lalu dihitung R^2 . Secara asimptotik, $nR^2 \sim \chi_p^2$. Implementasi uji tersebut di R:

```
bgtest(aovRAKL,order=1)
```

```
Breusch-Godfrey test for serial correlation of order up to 1
```

```
data: aovRAKL
LM test = 0.084631, df = 1, p-value = 0.7711
```

```
bgtest(aovRAKL,order=2)
```

Breusch-Godfrey test for serial correlation of order up to 2

```
data: aovRAKL
LM test = 7.5086, df = 2, p-value = 0.02342
```

```
bgtest(aovRAKL,order=4)
```

Breusch-Godfrey test for serial correlation of order up to 4

```
data: aovRAKL
LM test = 17.379, df = 4, p-value = 0.001631
```

Uji tersebut tak menolak H_0 di lag 1, tapi di lag selanjutnya uji tersebut menolak H_0 . Implementasi di Python:

```
from statsmodels.stats.diagnostic import acorr_breusch_godfrey
bgstat,pbg,fstat,pf=acorr_breusch_godfrey(RAKL_lm,nlags=2)
print("Statistik uji:", bgstat, "| P-value:", pbg)
```

Statistik uji: 7.508563644839594 | P-value: 0.023417262324381103

Diskusi mengenai kuasa uji autokorelasi dilakukan di [Ulyanto \(2020\)](#).

4.3.3 Konsekuensi dari ketidakbebasan pada galat

1. Ragam dari rata-rata contoh bukan lagi $(\frac{\sigma^2}{n})$ pada data yang tidak saling bebas

Ketika galat tidak saling bebas tapi sebaliknya memenuhi asumsi, maka dugaan dari pengaruh perlakuan yang kita peroleh masih tak bias, dan KTG mendekati tak bias untuk menduga σ^2 ketika ukuran contoh besar. Perbedaannya ialah ragam dari rata-rata tidak lagi σ^2 dibagi dengan ukuran sampel ($\frac{\sigma^2}{n}$). Ini berarti bahwa dugaan standar error untuk rata-rata perlakuan dan perbandingan perlakuan adalah dugaan yang bias, (apakah penyimpangannya terlalu besar atau kecil tergantung pada pola ketidak-salingbebasan pada galat), sehingga selang kepercayaan dan uji anova tidak akan memiliki laju kesalahan yang dibutuhkan. Uji-F pada ANOVA akan dipengaruhi oleh alasan yang sama.

2. Uji-F pada ANOVA kekar terhadap ketidakbebasan

Uji-F pada ANOVA kekar terhadap ketidakbebasan jika dipertimbangkan sebagai uji pengacakan. Artinya bahwa rata-rata di semua pengacakan yang mungkin, uji-F akan menolak hipotesis nol ketika hipotesis nol benar. Masalah yang lebih berat dapat muncul ketika tidak ada pengacakan pada seluruh ketidakbebasan. Misalnya, perlakuan mungkin diterapkan untuk unit-unit pada pengacakan. Tapi ketika respon-respon perlakuan diukur, diikuti oleh semua perlakuan dua unit, dan seterusnya. Penerapan pengacakan dari perlakuan untuk unit-unit tidak akan membantu, jika terdapat korelasi yang kuat pada pengukuran galat.

4.4 Aditivitas

Rancangan-rancangan seperti RAKL dan RBSL mengasumsikan bahwa pengaruh perlakuan dan kelompok **aditif** - pengaruh tersebut tetap sama di semua kombinasi perlakuan dan kelompok. Sifat tersebut terlihat dari model linear untuk RAKL:

$$y_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij}$$

Terlihat bahwa tidak ada interaksi ($\tau_i\beta_j$) antara perlakuan dan kelompok. Dalam kata lain, efek suatu kelompok dianggap sama di semua perlakuan. Jika asumsi tersebut tidak terpenuhi, dapat terlihat bahwa:

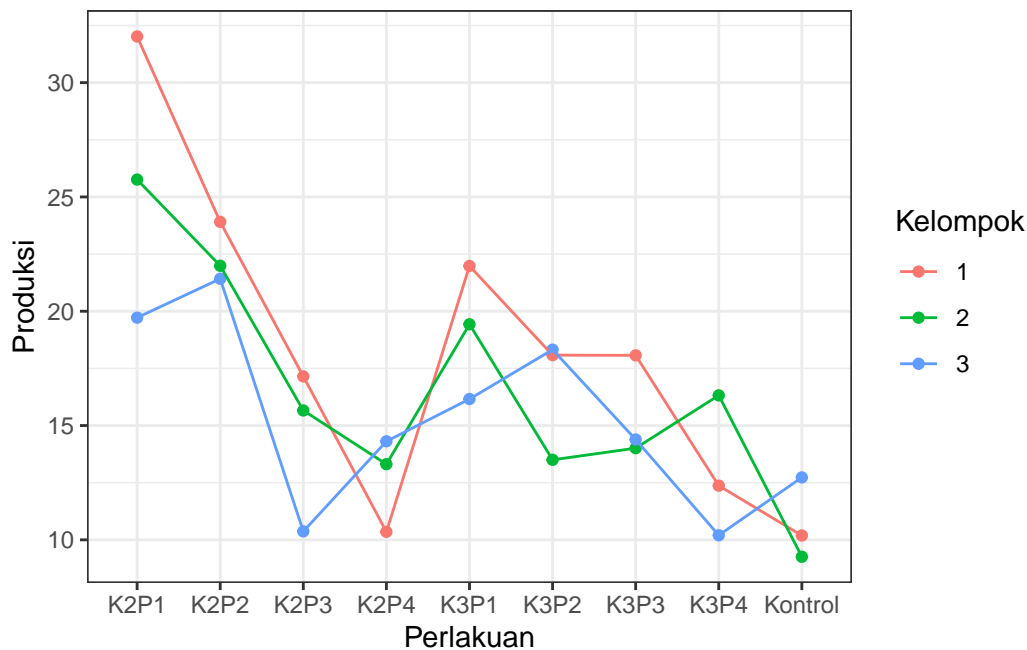
$$v_{ij} = \lambda_{ij} + \varepsilon_{ij} = y_{ij} - \mu - \tau_i - \beta_j$$

Akan menjadi komponen galat. Ini berarti galat akan heterogen tergantung besaran interaksi di kelompok dan perlakuan tertentu, sehingga ragam galat tidak terduga dengan benar. Oleh karena uji-F menggunakan penduga ragam galat, hasil-hasil uji dapat *misleading*. Ada beberapa cara menguji asumsi ini.

4.4.1 Plot Interaksi

Plot interaksi sudah didiskusikan di [Bab 2](#). Jika garis plot interaksi sejajar, asumsi keaditifan dianggap terpenuhi. Jika plot interaksi tidak sejajar, ada bukti untuk menduga bahwa asumsi keaditifan tidak terpenuhi. Sebagai contoh, plot interaksi ini:

```
library(ggplot2)
ggplot(DataRAKL, aes(x = Perlakuan, y = Produksi, colour = Kelompok)) +
  geom_point(data = DataRAKL, aes(y = Produksi)) +
  geom_line(data = DataRAKL, aes(y = Produksi, group = Kelompok)) +
  theme_bw()
```



Menandakan bahwa mungkin ada interaksi antara perlakuan dan kelompok karena garis plot interaksi tidak sejajar.

4.4.2 Uji formal

Asumsi aditivitas juga dapat diuji secara formal. Ada beberapa uji aditivitas yang dapat digunakan, antara lain uji Tukey, uji Mandel, uji Johnson-Graybill, uji Tussel, dan uji LBI (*locally best invariant*). Akan diberikan sintaks untuk tiap uji tersebut. Uji Tukey dan uji Tussel akan dijelaskan secara singkat. Filosofi inti dari uji Tukey mirip dengan uji Mandel,

sedangkan filosofi uji Johnson-Graybill, Tussel, dan LBI juga saling mirip. Oleh karena itu, hanya akan diberikan penjelasan singkat mengenai uji Tukey dan Tussel. Referensi tambahan mengenai uji aditivitas dapat dilihat di [Simeckova *et al.* \(2018\)](#)

4.4.2.1 Asumsi bentuk interaksi tertentu: uji Tukey

Jika kita menggunakan rancangan faktorial, di mana kombinasi perlakuan diulang sebanyak r kali, uji-F untuk efek interaksi merupakan bagian dari ANOVA. Namun, derajat bebas galat di rancangan faktorial adalah $ab(n - 1)$, sedangkan $n = 1$ di RAKL - satu perlakuan hanya muncul sekali di tiap kelompok. Derajat bebas galat akan nol jika kita menggunakan cara tersebut.

Namun, Uji Tukey pada dasarnya hanya menambahkan satu parameter θ . Parameter tersebut hanya mengambil satu derajat bebas sehingga derajat bebas galat tidak menjadi nol. Akan dibandingkan model tanpa interaksi dengan model berbentuk:

$$y_{ij} = \mu + \tau_i + \beta_j + \theta\tau_i\beta_j + \varepsilon_{ij}$$

Dalam kata lain, penduga interaksi dalam uji Tukey berbentuk $\lambda_{ij} = \theta\bar{y}_{i.}\bar{y}_{.j}$ - perkalian antara rata-rata perlakuan ke- i dan rata-rata kelompok ke- j , serta suatu konstanta. Oleh karena itu, hipotesis uji Tukey adalah:

$$H_0 : \theta = 0 \text{ (Model aditif)}$$

$$H_1 : \theta \neq 0 \text{ (Model tak aditif)}$$

Dengan statistik uji:

$$S_T = \frac{KT_{interaksi}}{KTG} \sim F_{(\alpha, 1, db_g)}$$

$$KT_{interaksi} = JK_{interaksi} = \frac{\left[\sum_{i=1}^t \sum_{j=1}^r y_{ij} (\bar{y}_{i.} - \bar{y}_{..}) (\bar{y}_{.j} - \bar{y}_{..}) \right]^2}{\sum_{i=1}^t (\bar{y}_{i.} - \bar{y}_{..})^2 \sum_{j=1}^r (\bar{y}_{.j} - \bar{y}_{..})^2}$$

$$JK_P = r \sum_{i=1}^t (\bar{y}_{i.} - \bar{y}_{..})^2$$

$$JK_K = t \sum_{j=1}^r (\bar{y}_{.j} - \bar{y}_{..})^2$$

$$JK_T = \sum_{i=1}^t \sum_{j=1}^r (\bar{y}_{ij} - \bar{y}_{..})^2$$

$$JK_G = JK_T - JK_P - JK_K - JK_{interaksi}$$

$$KT_g = \frac{JK_G}{db_g} = \frac{JK_G}{(t-1)(r-1) - 1}$$

Note, $db_g = (t-1)(r-1) - 1$ karena derajat bebas galat RAKL biasa berupa $(t-1)(r-1)$, tetapi satu derajat bebas diambil θ . Untuk mengimplementasikan uji di R, data harus berbentuk matriks - dalam kata lain, tiap kelompok merupakan baris, dan tiap kolom merupakan perlakuan. Lalu, kolom kelompok harus tidak ada. Matriks hanya berisi data, dengan tiap baris menandakan kelompok dan kolom menandakan perlakuan.

```
library(reshape2)
DataRAKLWide<-dcast(DataRAKL,Kelompok~Perlakuan,value.var="Produksi") # (data, baris~kolom)
Y<-as.matrix(DataRAKLWide[-1]) #keluarkan kolom 1 (nama kelompok)
```

Lakukan uji:

```
library(additivityTests)
tukey.test(Y,alpha=0.05) #alpha dapat dispesifikasi
```

Tukey test on 5% alpha-level:

Test statistic: 9.213

Critical value: 4.543

The additivity hypothesis was rejected.

Uji Tukey menolak H_0 , sehingga dapat disimpulkan $\theta \neq 0$ dan model tak aditif. Bandingkan dengan uji Mandel:

```
mandel.test(Y,alpha=0.05)
```

Mandel test on 5% alpha-level:

Test statistic: 4.316

Critical value: 3.739

The additivity hypothesis was rejected.

Uji tersebut juga menolak H_0 .

4.4.2.2 Uji dari akar ciri: Tussel, LBI, Johnson-Graybill

Uji Tukey mengasumsikan bentuk interaksi tertentu. Namun, bisa saja bentuk interaksi antara kelompok dan perlakuan lebih kompleks dari yang diberikan uji Tukey. Dalam kasus ini, kuasa uji (probabilitas menolak H_0 jika H_1 benar) dari uji Tukey dan Mandel relatif lebih rendah.

Ketiga uji ini muncul dari mengkaji matriks residual R , di mana elemen matriks tersebut adalah:

$$r_{ij} = y_{ij} - \bar{y}_{i.} - \bar{y}_{.j} + \bar{y}_{..}$$

Lalu, matriks tersebut akan dikali dengan transposenya, RR^T dan akan dicari akar ciri matriks tersebut. Akar ciri diurutkan sehingga $k_1 \geq k_2 \geq \dots \geq k_n$. Lalu, tiap akar ciri dibagi dengan jumlah total akar ciri, sehingga $\omega_i = k_i / \sum_{i=1}^n k_i$. Jika ada interaksi, beberapa ω_i akan lebih besar dari yang lainnya. Statistik uji Johnson-Graybill merupakan akar ciri terstandarisasi terbesar ω_1 . Statistik uji Tussel dan LBI mencari m jumlah akar ciri, di mana jumlah tersebut adalah $\min(r, t) - 1$. Maka, dicari jumlah yang terkecil dari jumlah perlakuan atau jumlah kelompok. Angka tersebut dikurangi satu dan diambil m akar ciri sebanyak jumlah tersebut. LBI menambahkan akar ciri tersebut sedangkan Tussel mengalikan akar ciri. Johnson Graybill dan LBI menolak H_0 jika statistik ujinya lebih besar dari nilai kritis. Uji Tussel menolak H_0 jika statistik ujinya lebih kecil dari nilai kritis. Implementasi uji di R:

```
johnson.graybill.test(Y)
```

Johnson and Graybill test on 5% alpha-level:

Test statistic: 0.7262

Critical value: 0.8799

The additivity hypothesis cannot be rejected.

```
lbi.test(Y)
```

Locally Best Invariant test on 5% alpha-level:

Test statistic: 0.6024

Critical value: 0.7797

The additivity hypothesis cannot be rejected.

```
tusell.test(Y)
```

Tusell test on 5% alpha-level:

Test statistic: 0.1988

Critical value: 0.1089

The additivity hypothesis cannot be rejected.

Dapat dilihat bahwa hasil uji Tukey dan Mandel berbeda dengan uji berdasarkan akar ciri. α sama - dalam kata lain, probabilitas menolak H_0 jika H_0 benar sama. Namun, kuasa uji berbeda. Jika bentuk interaksi sederhana ($\theta\tau_i\beta_j$), uji Tukey dan Mandel memiliki kuasa lebih besar dari uji berdasarkan akar ciri. Namun, jika bentuk interaksi lebih kompleks, uji akar ciri memiliki kuasa lebih besar.

Dalam kasus ini, dapat dilihat skenario-skenario tertentu. Jika interaksi kompleks, seharusnya uji Tukey dan Mandel cenderung taktolak H_0 (karena kuasa uji lebih rendah) dan uji akar ciri cenderung menolak H_0 . Jika interaksi sederhana, seharusnya uji Tukey dan Mandel cenderung tolak H_0 (karena kuasa uji lebih tinggi) dan uji akar ciri cenderung taktolak H_0 . Jika tidak ada interaksi, α sudah dibuat sama sehingga seharusnya tidak ada kecenderungan tertentu.

Kasus ini sepertinya sesuai dengan skenario interaksi sederhana. Oleh karena itu, hasil uji Tukey dan Mandel dapat lebih dipercaya. Selain itu, sudah ada plot interaksi yang memperkuat kepercayaan bahwa interaksi benar adanya.

Sebagai tambahan terakhir, implementasi uji Tukey di Python dapat dilihat di notebook Google Colab [ini](#).

5 Rancangan Faktorial

Di kebanyakan aplikasi percangan percobaan, ada lebih dari satu faktor yang memengaruhi respon tertentu. Misal, produksi suatu petak tanah ditentukan oleh pemupukan, varietas yang ditanam, irigasi, dan lain-lain. Bahkan, pengujian pengaruh satu faktor belum tentu valid jika level faktor-faktor lain berbeda. Misal, dalam kondisi irigasi yang cukup, varietas yang berbeda akan memiliki produktivitas berbeda. Namun, irigasi yang tidak cukup akan menyebabkan semua varietas mati kekurangan air sehingga produksi semua varietas sama-sama nol. Contoh tersebut cukup ekstrim, tetapi intinya adalah respon merupakan akibat dari berbagai faktor secara simultan, tidak terpisah (**interaksi**). Belum tentu, misal, tambahan pemupukan 1 *kg* selalu menambah produktivitas 1 *kg/ha* - efek tersebut mungkin lebih besar di kondisi tertentu (varietas yang reseptif pada pupuk, irigasi baik). Oleh karena itu, perlu rancangan percobaan dengan lebih dari satu faktor untuk beberapa aplikasi tertentu. Selain itu, semua **kombinasi** taraf faktor harus dicobakan, karena pengaruh taraf suatu faktor dapat berbeda-beda jika dipasangkan dengan taraf faktor lain yang berbeda.

Note, hal ini berarti faktor-faktor harus bersilang, tidak bersarang. Jika level faktor tertentu hanya muncul saat faktor memiliki level tertentu juga, tidak dapat diuji semua kombinasi perlakuan. Jika saat kerapatan tanaman tinggi, misal, varietas tertentu tidak dapat ditanam, tidak mungkin dilakukan rancangan faktorial.

5.1 Faktorial RAL

5.1.1 Pembuatan bagan percobaan

Karena perlakuan di suatu percobaan faktorial merupakan semua kombinasi taraf faktor yang diteliti, pembuatan bagan percobaan diawali dengan penjabaran taraf-taraf faktor tersebut. Lalu digunakan fungsi `expand.grid` dari package `data.table` untuk mengkombinasikan taraf-taraf faktor:

```
library(data.table)
trt1<-c("A","B","C")
trt2<-c("D","E","F")
perlakuan<-data.frame(expand.grid(trt1,trt2)) #buat kombinasi
colnames(perlakuan)<-c("Faktor 1", "Faktor 2") #penamaan faktor, berupa kosmetik
```

```
knitr::kable(perlakuan)
```

Faktor 1	Faktor 2
A	D
B	D
C	D
A	E
B	E
C	E
A	F
B	F
C	F

Terlihat bahwa tiap baris merupakan kombinasi unik dari taraf faktor satu dan dua. Jumlah baris juga tepat, yaitu $3 \times 3 = 9$. Lalu, kode bagi tiap kombinasi tersebut dapat diekstraksi dan dibuat. Tiap kombinasi dapat dikodekan dengan angka, atau dengan teks yang merupakan gabungan dari nama taraf faktor pertama dan kedua.

```
perlakuan$kode<-seq(1,nrow(perlakuan)) #angka
perlakuan$kode2<-paste(perlakuan$`Faktor 1`,perlakuan$`Faktor 2`,sep="") #gabungan nama ta
knitr::kable(perlakuan)
```

Faktor 1	Faktor 2	kode	kode2
A	D	1	AD
B	D	2	BD
C	D	3	CD
A	E	4	AE
B	E	5	BE
C	E	6	CE
A	F	7	AF
B	F	8	BF
C	F	9	CF

Lalu, buat saja bagan tersebut melalui design.crd (RAL) di library agricolae. Masukkan jumlah ulangan yang diinginkan juga. Akan ditunjukkan sepuluh unit percobaan pertama sebagai ilustrasi:

```
library(agricolae)
baganFRAL<-design.crd(trt=perlakuan$kode2,r=4,seed=16,serie=0)

#akses output -> hasil design.crd$book
knitr::kable(head(baganFRAL$book,n=10))
```

plots	r	perlakuan\$kode2
1	1	AF
2	1	CD
3	1	BD
4	1	CE
5	1	CF
6	1	AE
7	2	AF
8	3	AF
9	2	CD
10	2	CE

Dapat diverifikasi bahwa jumlah unit percobaan merupakan $9 \times 4 = 36$.

```
nrow(baganFRAL$book)
```

```
[1] 36
```

5.1.1.1 Pengacakan edible

Karena rancangan pengendalian lingkungan berupa RAL, hanya ada satu jenis unit. Lalu, ada dua perlakuan di `set_trts` yang semuanya dialokasikan ke unit secara bersamaan. Oleh karena itu, tulis `c(P1.,P2.)` untuk menggabungkan kedua perlakuan, lalu `allot_trts(c(P1.,P2.)~unit)` untuk mengalokasikan semua kombinasi perlakuan tersebut ke unit.

```
library(edibble)

desFaktRAL<-design(name="Faktorial RAL") %>%
  set_units(unit=36) %>%
  set_trts(P1.=trt1,
           P2.=trt2) %>%
  allot_trts(c(P1.,P2.) ~unit) %>%
```

```
assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desFaktRAL,n=10))
```

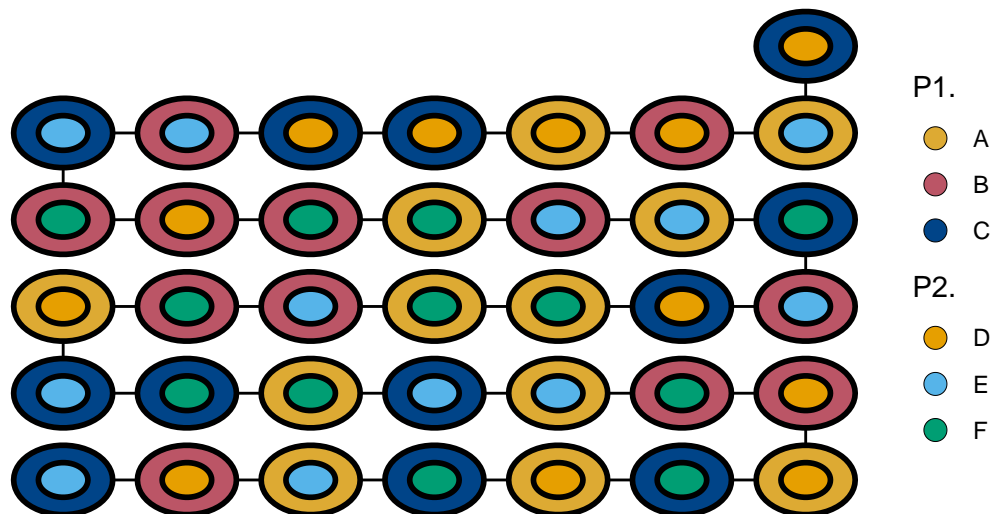
unit	P1.	P2.
unit1	C	E
unit2	B	D
unit3	A	E
unit4	C	F
unit5	A	D
unit6	C	F
unit7	A	D
unit8	B	D
unit9	B	F
unit10	A	E

Lalu, plot-kan:

```
deggust::autoplot(desFaktRAL)
```

Faktorial RAL

Unit: unit



Bandingkan dengan default edible:

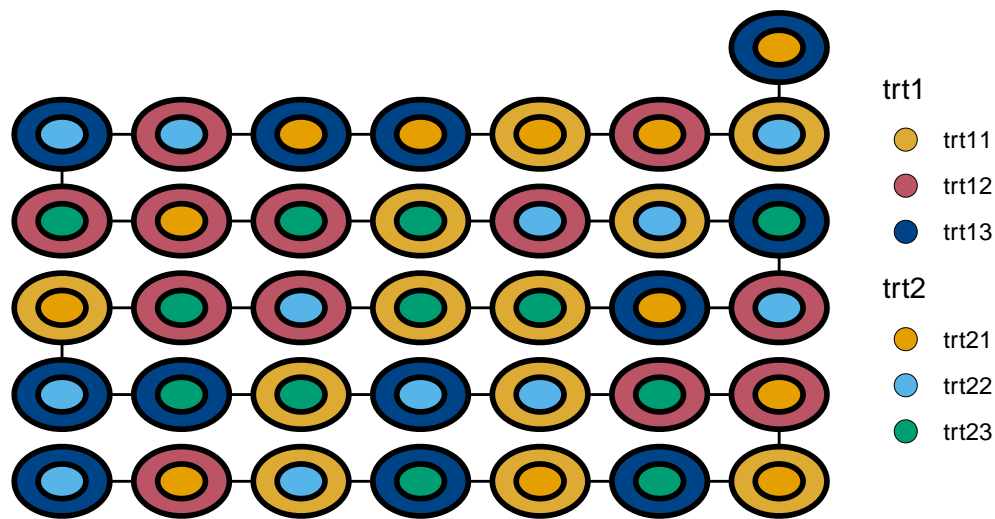
```
fac <- takeout(menu_factorial(trt = c(3, 3), r=4, seed=420))
examine_recipe(fac)
```

```
design("Factorial Design") %>%
  set_units(unit = 36) %>%
  set_trts(trt1 = 3,
           trt2 = 3) %>%
  allot_trts(~unit) %>%
  assign_trts("random", seed = 420) %>%
  serve_table()
```

```
deggust::autoplot(fac)
```

Factorial Design

Unit: unit



Cara pembuatan dan plot sama di kedua kasus.

5.1.2 Model Linear Aditif

Model bagi faktorial RAL adalah sebagai berikut:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

Dengan:

1. Y_{ijk} adalah nilai pengamatan pada faktor A taraf ke-i dan faktor B taraf ke-j pada ulangan ke-k.
2. μ adalah komponen aditif dari rata-rata umum.
3. α_i adalah komponen aditif dari pengaruh faktor A pada taraf ke-i.
4. β_j adalah komponen aditif dari pengaruh faktor B pada taraf ke-j.
5. $(\alpha\beta)_{ij}$ adalah komponen interaksi faktor A dan faktor B.
6. $\varepsilon_{ijk} \sim N(0, \sigma_\varepsilon^2)$ adalah komponen acak pada tiap pengamatan

Oleh karena itu, terdapat tiga hipotesis yang diuji, yaitu hipotesis mengenai pengaruh faktor A, pengaruh faktor B, dan pengaruh interaksi:

Pengaruh faktor A:

$$H_0 : \alpha_1 = \dots = \alpha_a = 0$$

(faktor A tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada satu } i \text{ di mana } \alpha_i \neq 0$$

Pengaruh faktor B:

$$H_0 : \beta_1 = \dots = \beta_b = 0$$

(faktor B tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada satu } j \text{ di mana } \beta_j \neq 0$$

Pengaruh interaksi:

$$H_0 : (\alpha\beta)_{11} = (\alpha\beta)_{12} = \dots = (\alpha\beta)_{ab} = 0$$

(Interaksi faktor A dan B tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada sepasang } (i, j) \text{ di mana } (\alpha\beta)_{ij} \neq 0$$

Hipotesis tersebut diuji dengan uji-F, yang disimpulkan di tabel sidik ragam:

```
cat('
| Sumber Keragaman| db| JK| KT| F-hit|F(dbP,dbG)|
|-----:|-----:|-----:|-----:|-----:|
|          A|          a-1|          JKA| JKP/dbA | KTA/KTG | |
|          B|          b-1|          JKB| JKB/dbB | KTB/KTG | |
|          A*B|          (a-1)(b-1)|          JKAB| JKAB/dbAB | KTAB/KTG | |
|          Galat|          ab(r-1)|          JKG| JKG/dbG | | |
```

	Total	abr-1	JKT			')
--	-------	-------	-----	--	--	----

	Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
	-----:	-----:	-----:	-----:	-----:	-----:
	A	a-1	JKA	JKP/dbA	KTA/KTG	
	B	b-1	JKB	JKB/dbB	KTB/KTG	
	A*B	(a-1)(b-1)	JKAB	JKAB/dbAB	KTAB/KTG	
	Galat	ab(r-1)	JKG	JKG/dbG		
	Total	abr-1	JKT			

Di mana jumlah kuadrat tersebut dihitung dengan:

$$\begin{aligned}
 FK &= \frac{y_{...}^2}{abr} \\
 JKT &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (y_{ijk} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r y_{ijk}^2 - FK \\
 JKA &= \sum_{i=1}^a (\bar{y}_{i..} - \bar{y}_{...})^2 = \sum_{i=1}^a \frac{y_{i..}^2}{br} - FK \\
 JKB &= \sum_{j=1}^b (\bar{y}_{.j.} - \bar{y}_{...})^2 = \sum_{j=1}^b \frac{y_{.j.}^2}{ar} - FK \\
 JKP &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \frac{y_{ij.}^2}{r} - FK \\
 JKAB &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 - JKA - JKB \\
 &= JKP - JKB - JKA \\
 JKG &= JKP - JKT
 \end{aligned}$$

5.1.3 ANOVA dengan R

Akan digunakan kasus berikut dari Mattjik dan Sumertajaya:

Balai Karantina ingin mengetahui pengaruh pemberian fumigasi dengan berbagai dosis (0, 16, 32, 48, 64; g/m^3) dengan lama fumigasi yang berbeda (2 dan 4 jam) terhadap daya kecambah benih tomat. Metode perkecambahan yang digunakan adalah *Growing on Test*. Unit percobaan diasumsikan homogen.

Pertama, baca data terlebih dahulu:

```
library(google sheets4)
```

```
Fact<-read_sheet("https://docs.google.com/spreadsheets/d/1HuYXEBH1pJcXY2v-XJfmiUoFSN05p1U  
knitr::kable(Fact)
```

LamaFumigasi	Ulangan	Dosis Fumigan	...4	...5	...6	...7
NA	NA	0	16	32	48	64
2	1	96	92	92	74	50
2	2	98	88	94	74	50
2	3	94	90	84	68	54
4	1	90	88	78	0	0
4	2	94	92	82	0	0
4	3	92	94	74	0	0

Terlihat bahwa baris pertama merupakan taraf-taraf dari faktor dosis. Oleh karena itu, baris tersebut sebaiknya dihilangkan. Selain itu, kolom ketiga sampai terakhir hendak diberi nama sesuai dosis diberikan.

```
Fact<-read_sheet("https://docs.google.com/spreadsheets/d/1HuYXEBH1pJcXY2v-XJfmiUoFSN05p1U  
colnames(Fact)[seq(3,ncol(Fact))]<-seq(0,64,16)
```

Lalu, data tersebut dirubah ke format *long* di mana tiap baris menunjukkan unit percobaan individu, bukan kombinasi nama ulangan dan dosis fumigasi. Seperti biasa, *id.vars* merupakan pembeda individu di format *wide* awal (dalam kasus ini, tiap individu adalah kombinasi LamaFumigasi dan Ulangan). Lalu, *value.vars* merupakan kolom-kolom yang digabung menjadi satu, dalam kasus ini dosis-dosis dari 0 sampai 64.

```
library(reshape2)
```

```
Fact2<-melt(Fact,  
  #variabel yang membedakan tiap baris di tabulasi asli:  
  id.vars=c("LamaFumigasi","Ulangan"),  
  value.vars=seq(0,64,16),  
  value.name="Perkecambahan")  
colnames(Fact2)[3]<-"Dosis" #kolom ketiga akan disebut "variable"
```

Setelah tabulasi data benar, pastikan tiap faktor sudah dibuat menjadi objek data yang benar; lakukan ANOVA dan lihat hasil:

```
Fact2$LamaFumigasi<-as.factor(Fact2$LamaFumigasi)
Fact2$Dosis<-as.factor(Fact2$Dosis)
aovFact<-aov(Perkecambahan~LamaFumigasi+Dosis+LamaFumigasi:Dosis,Fact2)
summary(aovFact)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
LamaFumigasi	1	5713	5713	691.1	< 2e-16 ***
Dosis	4	25459	6365	769.9	< 2e-16 ***
LamaFumigasi:Dosis	4	6258	1565	189.3	1.37e-15 ***
Residuals	20	165	8		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Analisis di python dapat dilakukan dengan membaca CSV dan mengambil row ke-2 sampai akhir. Note bahwa indeks di Python dimulai dari 0, sehingga baris ke-2 dinotasikan dengan 1.

```
import pandas as pd
```

```
fum=pd.read_csv("https://docs.google.com/spreadsheets/d/1HuYXEBHlpJcXY2v-XJfmiUoFSN05p1ULD")
fumn=fum.iloc[1:]
```

Lalu, ubah nama kolom:

```
print(fumn.columns)
```

```
Index(['LamaFumigasi', 'Ulangan', 'Dosis Fumigan', 'Unnamed: 3', 'Unnamed: 4',
      'Unnamed: 5', 'Unnamed: 6'],
      dtype='object')
```

Ambil elemen pertama dan kedua (0, 1) - indexing di Python dimulai dari 0 dan berakhir di elemen $n - 1$ sehingga ambil elemen [0:2]. Elemen tersebut diubah menjadi list. Lalu tambahkan angka 0, 16, 32, 48, 64 kepada list tersebut. Lalu masukkan list tersebut ke kolom-kolom dataset:

```
fumn.columns=list(fumn.columns[0:2])+[str(x) for x in range(0,80,16)]
fumn.head()
```

```
LamaFumigasi  Ulangan    0   16   32   48   64
```

1	2.0	1.0	96	92	92	74	50
2	2.0	2.0	98	88	94	74	50
3	2.0	3.0	94	90	84	68	54
4	4.0	1.0	90	88	78	0	0
5	4.0	2.0	94	92	82	0	0

Melt data:

```
FaktRALmelt=pd.melt(fumn, id_vars=['LamaFumigasi','Ulangan'], value_vars=[str(x) for x in
```

Lakukan ANOVA. Gunakan * untuk mendapat interaksi, dan C() untuk membuat peubah kategorik:

```
import statsmodels.api as sm
```

```
C:\Users\Acer\AppData\Local\R\win-library\4.2\reticulate\python\rpytools\loader.py:39: FutureWarning:
    module = _import(
C:\Users\Acer\AppData\Local\Programs\Python\PYTHON~2\lib\site-packages\statsmodels\compat\pandas.py:10:
    from pandas import Int64Index as NumericIndex
```

```
from statsmodels.formula.api import ols
```

```
FaktRALlm= ols('value ~ C(LamaFumigasi)*C(variable)',data=FaktRALmelt)
fitRAL=FaktRALlm.fit()
```

```
table = sm.stats.anova_lm(fitRAL)
print(table)
```

	df	sum_sq	...	F	PR(>F)
C(LamaFumigasi)	1.0	5713.200000	...	691.112903	5.527620e-17
C(variable)	4.0	25459.200000	...	769.935484	1.367358e-21
C(LamaFumigasi):C(variable)	4.0	6258.133333	...	189.258065	1.370943e-15
Residual	20.0	165.333333	...	NaN	NaN

[4 rows x 5 columns]

5.1.4 Tabel dan Plot Interaksi

Sekarang, dapat dicari rata-rata pengaruh perlakuan di tiap kombinasi faktor A dan B. Fungsi yang akan digunakan adalah aggregate dari base R. Argumen pertama merupakan fungsi

agregasi yang ingin dilakukan - sintaks ini mirip dengan aov atau lm (respon~perlakuan), argumen kedua merupakan faktor, dan argumen terakhir merupakan fungsi agregasi yang dipakai (dalam kasus ini, mean):

```
knitr::kable(aggregate(Perkecambahan~Dosis+LamaFumigasi,data=Fact2,FUN=mean))
```

Dosis	LamaFumigasi	Perkecambahan
0	2	96.00000
16	2	90.00000
32	2	90.00000
48	2	72.00000
64	2	51.33333
0	4	92.00000
16	4	91.33333
32	4	78.00000
48	4	0.00000
64	4	0.00000

Lalu, buat tiap baris menjadi taraf faktor A (Lama Fumigasi), dan tiap kolom menjadi taraf faktor B. Gunakan fungsi dcast, dengan sintaks dcast(tabel, fungsi: baris~kolom, peubah respon).

```
SumTab<-reshape2::dcast(aggregate(Perkecambahan~Dosis+LamaFumigasi,data=Fact2,FUN=mean),LamaFumigasi~Dosis)
knitr::kable(SumTab)
```

LamaFumigasi	0	16	32	48	64
2	96	90.00000	90	72	51.33333
4	92	91.33333	78	0	0.00000

Lalu, tambahkan rerata tiap baris (rowMeans). Namun, kolom pertama (lama fumigasi) tidak merupakan peubah respon dan tidak diperhitungkan:

```
SumTab$Rerata<-rowMeans(SumTab[, -1])
knitr::kable(SumTab)
```

LamaFumigasi	0	16	32	48	64	Rerata
2	96	90.00000	90	72	51.33333	79.86667

LamaFumigasi	0	16	32	48	64	Rerata
4	92	91.33333	78	0	0.00000	52.26667

Lalu, tambahkan rerata per kolom:

```
SumTab<-rbind(SumTab,colMeans(SumTab[,-1]))
```

Warning in rbind(deparse.level, ...): number of columns of result, 7, is not a multiple of vector length 6 of arg 2

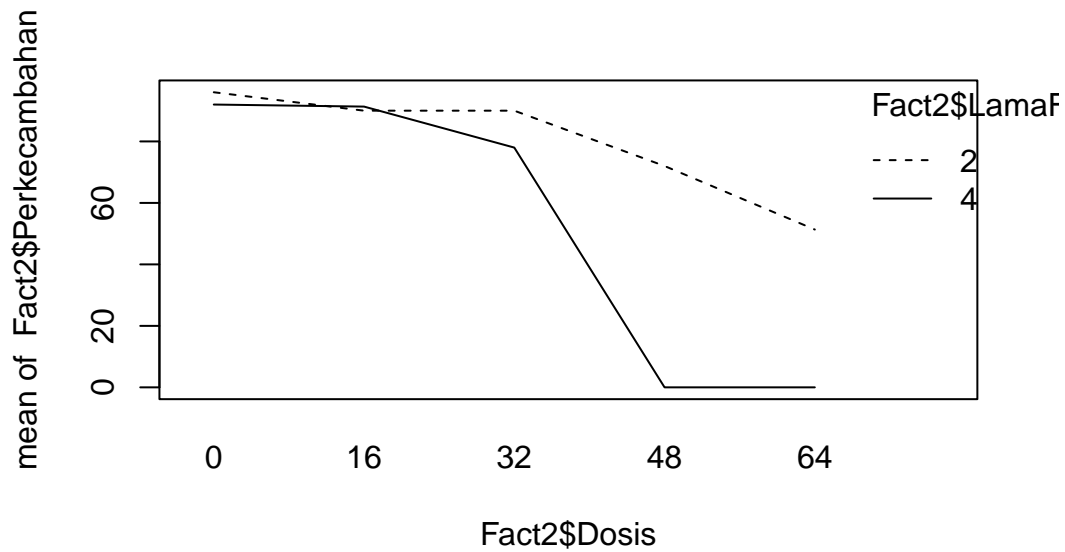
Warning in `[<-.factor`(`*tmp*`, ri, value = 94): invalid factor level, NA generated

```
knitr::kable(SumTab)
```

LamaFumigasi	0	16	32	48	64	Rerata
2	96.00000	90.00000	90	72.00000	51.33333	79.86667
4	92.00000	91.33333	78	0.00000	0.00000	52.26667
NA	90.66667	84.00000	36	25.66667	66.06667	94.00000

Tabel rata-rata perlakuan telah terbentuk. Tabel ini dapat digunakan untuk pembuatan plot interaksi secara manual. Di R, plot interaksi dibuat dengan:

```
interaction.plot(Fact2$Dosis, Fact2$LamaFumigasi, Fact2$Perkecambahan) #(faktor 1, faktor
```

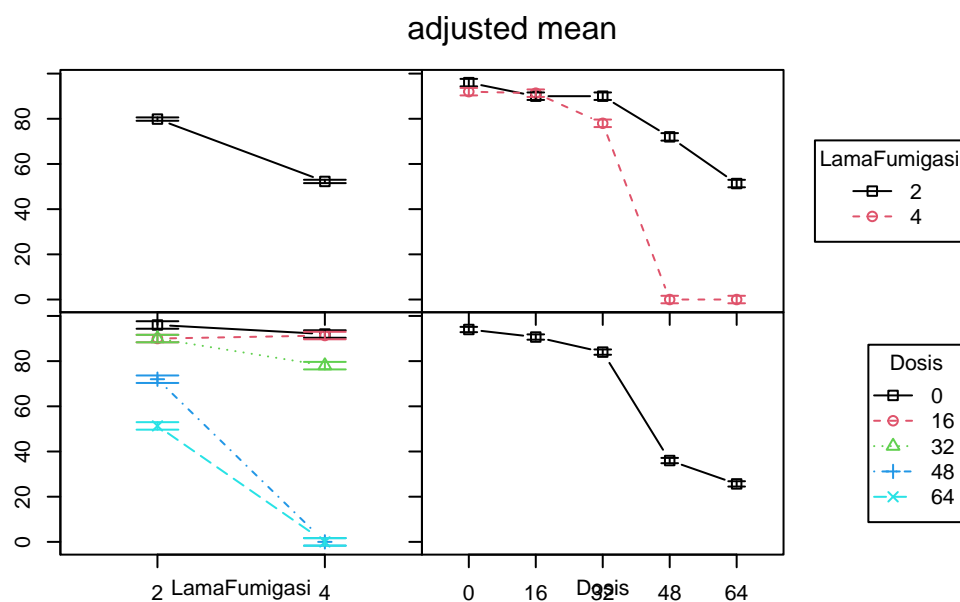


Atau, dapat dibuat plot interaksi melalui library phia. Library phia hanya menerima input berupa lm - ubah olah data menggunakan lm terlebih dahulu. Phia menyediakan tabel rata-rata perlakuan, serta dapat di-plotkan menjadi plot interaksi.

```
library(phia)
mod<-lm(Perkecambahan~LamaFumigasi+Dosis+LamaFumigasi:Dosis,data=Fact2)
im=interactionMeans(mod)
knitr::kable(im) # tabel
```

LamaFumigasi	Dosis	adjusted mean	std. error
2	0	96.00000	1.659987
4	0	92.00000	1.659987
2	16	90.00000	1.659987
4	16	91.33333	1.659987
2	32	90.00000	1.659987
4	32	78.00000	1.659987
2	48	72.00000	1.659987
4	48	0.00000	1.659987
2	64	51.33333	1.659987
4	64	0.00000	1.659987


```
plot(im) # plot
```



5.1.5 Uji lanjut

Uji lanjut dapat dilakukan dengan library emmeans. Pada dasarnya, emmeans menghasilkan output yang mirip dengan uji lanjut seperti TukeyHSD:

```
library(emmeans)
marginal = emmeans(mod, ~ LamaFumigasi:Dosis)
knitr::kable(head(pairs(marginal, adjust="Tukey"), n=5))
```

contrast	estimate	SE	df	t.ratio	p.value
LamaFumigasi2 Dosis0 - LamaFumigasi4 Dosis0	4	2.347576	20	1.703885	0.1038913

```
knitr::kable(head(TukeyHSD(aovFact, conf.level=.95)$`LamaFumigasi:Dosis`), n=5) #metode tuk
```

	diff	lwr	upr	p adj
4:0-2:0	-4.000000	-12.31302	4.313018	0.7813125
2:16-2:0	-6.000000	-14.31302	2.313018	0.2987279
4:16-2:0	-4.666667	-12.97968	3.646352	0.6158480
2:32-2:0	-6.000000	-14.31302	2.313018	0.2987279
4:32-2:0	-18.000000	-26.31302	-9.686982	0.0000082
2:48-2:0	-24.000000	-32.31302	-15.686982	0.0000001

Urutan pengurangnya berbeda saja, sehingga hasil emmeans memiliki beda positif dan TukeyHSD memiliki beda negatif. Library multcomp memiliki output yang lebih rapi, yaitu langsung berupa pengelompokan:

```
library(multcomp)
```

```
Loading required package: mvtnorm
```

```
Loading required package: survival
```

```
Loading required package: TH.data
```

```
Loading required package: MASS
```

```
Attaching package: 'TH.data'
```

```
The following object is masked from 'package:MASS':
```

```
geyser
```

```
knitr::kable(cld(marginal,
alpha=0.05,
Letters=letters,
adjust="tukey"))
```

Note: adjust = "tukey" was changed to "sidak" because "tukey" is only appropriate for one set of pairwise comparisons

	LamaFumigasi	Dosis	emmean	SE	df	lower.CL	upper.CL	.group
8	4	48	0.00000	1.659987	20	-5.21784	5.21784	a
10	4	64	0.00000	1.659987	20	-5.21784	5.21784	a
9	2	64	51.33333	1.659987	20	46.11549	56.55117	b
7	2	48	72.00000	1.659987	20	66.78216	77.21784	c
6	4	32	78.00000	1.659987	20	72.78216	83.21784	c
3	2	16	90.00000	1.659987	20	84.78216	95.21784	d
5	2	32	90.00000	1.659987	20	84.78216	95.21784	d
4	4	16	91.33333	1.659987	20	86.11549	96.55117	d
2	4	0	92.00000	1.659987	20	86.78216	97.21784	d
1	2	0	96.00000	1.659987	20	90.78216	101.21784	d

Selain itu, dapat diuji kontras orthogonal dari data ini. Karena hanya ada dua taraf untuk lama fumigasi, kontras hanya dapat dibuat untuk dosis. Misal dibuat kontras untuk membandingkan pengaruh ada dosis (dosis 0 vs lainnya), dosis rendah vs tinggi (16,32 vs 48,64), serta perbandingan antara dosis 16 vs 32 dan 48 vs 64:

```
levels(Fact2$Dosis)
```

```
[1] "0" "16" "32" "48" "64"
```

```
contrasts(Fact2$Dosis)<- cbind(c(1, -1/4, -1/4, -1/4,-1/4), c(0, 1/2,1/2, -1/2, -1/2),
                               c(0, 1,-1, 0, 0), c(0,0,0,1,-1))
contrasts(Fact2$Dosis)
```

```
      [,1] [,2] [,3] [,4]
0      1.00  0.0   0    0
16     -0.25  0.5   1    0
32     -0.25  0.5  -1    0
48     -0.25 -0.5   0    1
64     -0.25 -0.5   0   -1
```

```
aovFact1<-aov(Perkecambahan~LamaFumigasi+Dosis+LamaFumigasi:Dosis,Fact2)
summary.aov(aovFact1,split=list(Dosis=list("0 vs ada"=1,"16,32 vs 48,64"=2,"16 vs 32"=3,"48 vs 64"=4)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
LamaFumigasi	1	5713	5713	691.11	< 2e-16 ***
Dosis	4	25459	6365	769.93	< 2e-16 ***

Dosis: 0 vs ada	1	5852	5852	707.91	< 2e-16	***
Dosis: 16,32 vs 48,64	1	19154	19154	2316.96	< 2e-16	***
Dosis: 16 vs 32	1	133	133	16.13	0.000678	***
Dosis: 48 vs 64	1	320	320	38.75	4.43e-06	***
LamaFumigasi:Dosis	4	6258	1565	189.26	1.37e-15	***
LamaFumigasi:Dosis: 0 vs ada	1	1044	1044	126.33	4.28e-10	***
LamaFumigasi:Dosis: 16,32 vs 48,64	1	4760	4760	575.83	3.25e-16	***
LamaFumigasi:Dosis: 16 vs 32	1	133	133	16.13	0.000678	***
LamaFumigasi:Dosis: 48 vs 64	1	320	320	38.75	4.43e-06	***
Residuals	20	165	8			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Dapat juga dibuat polinomial orthogonal, dengan koefisien-koefisien sesuai yang ditentukan saat taraf faktor tersebut 4.

```
contrasts(Fact2$Dosis) <- cbind(c(-2, -1, 0, 1, 2), c(2, -1, -2, -1, 2),
                                c(-1, 2, 0, -2, 1), c(1, -4, 6, -4, 1))
contrasts(Fact2$Dosis)
```

	[,1]	[,2]	[,3]	[,4]
0	-2	2	-1	1
16	-1	-1	2	-4
32	0	-2	0	6
48	1	-1	-2	-4
64	2	2	1	1

```
aovFact2<-aov(Perkecambahan~LamaFumigasi+Dosis+LamaFumigasi:Dosis,Fact2)
summary.aov(aovFact2,split=list(Dosis=list("Linear"=1,"Kuadrat"=2,"Kubik"=3,"Kuartik"=4))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
LamaFumigasi	1	5713	5713	691.113	< 2e-16 ***
Dosis	4	25459	6365	769.935	< 2e-16 ***
Dosis: Linear	1	21965	21965	2657.065	< 2e-16 ***
Dosis: Kuadrat	1	1312	1312	158.733	5.72e-11 ***
Dosis: Kubik	1	1009	1009	122.008	5.79e-10 ***
Dosis: Kuartik	1	1173	1173	141.937	1.54e-10 ***
LamaFumigasi:Dosis	4	6258	1565	189.258	1.37e-15 ***
LamaFumigasi:Dosis: Linear	1	4234	4234	512.129	1.01e-15 ***
LamaFumigasi:Dosis: Kuadrat	1	27	27	3.318	0.0835 .

```

LamaFumigasi:Dosis: Kubik      1    1480    1480    179.040 1.94e-11 ***
LamaFumigasi:Dosis: Kuartik    1     517     517     62.545 1.39e-07 ***
Residuals                      20     165        8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

5.2 Faktorial RAKL

Jika kondisi unit percobaan tidak sepenuhnya homogen, dapat digunakan Faktorial RAKL. Sama seperti RAKL pada kasus satu faktor, tiap perlakuan diacak dan diulang sekali di tiap kelompok.

5.2.1 Pengacakan

Pertama, buat tiap kombinasi perlakuan:

```

library(data.table)

vari<-c("V1","V2","V3")
pupu<-c("N0","N1","N2","N3")
KLperlakuan<-expand.grid(vari,pupu) #buat kombinasi
colnames(KLperlakuan)<-c("Varietas","Pupuk") #ubah nama kolom
knitr::kable(KLperlakuan)

```

Varietas	Pupuk
V1	N0
V2	N0
V3	N0
V1	N1
V2	N1
V3	N1
V1	N2
V2	N2
V3	N2
V1	N3
V2	N3
V3	N3

Lalu, beri penomoran bagi tiap kombinasi perlakuan tersebut:

```
KLperlakuan$Nomor<-paste(KLperlakuan$Varietas,KLperlakuan$Pupuk,sep="")
knitr::kable(KLperlakuan)
```

Varietas	Pupuk	Nomor
V1	N0	V1N0
V2	N0	V2N0
V3	N0	V3N0
V1	N1	V1N1
V2	N1	V2N1
V3	N1	V3N1
V1	N2	V1N2
V2	N2	V2N2
V3	N2	V3N2
V1	N3	V1N3
V2	N3	V2N3
V3	N3	V3N3

Dari tiap perlakuan tersebut, lakukan pengacakan di tiap kelompok. Di R, ini dilakukan dengan `design.rcbd` dari `agricolae`:

```
library(agricolae)

baganFakt_RAKL<-design.rcbd(KLperlakuan$Nomor,3,seed=78) #kombinasi perlakuan, jml kelompok
knitr::kable(baganFakt_RAKL$sketch)
```

V1N3	V2N3	V2N2	V2N0	V2N1	V1N1	V3N0	V3N2	V3N1	V1N0	V3N3	V1N2
V2N3	V3N2	V1N0	V1N1	V3N0	V1N3	V2N2	V2N1	V3N1	V1N2	V3N3	V2N0
V3N0	V3N2	V2N3	V2N1	V2N2	V1N0	V2N0	V1N2	V3N1	V1N3	V3N3	V1N1

5.2.1.1 Pengacakan edibble

Pada dasarnya, struktur perlakuan akan sama, tetapi struktur unit dibuat sesuai dengan RAKL, yaitu membuat unit kelompok dan petak yang `nested_in(kelompok)`. Perlakuan masih diacak ke petak (unit):

```
desFaktRAKL<-design(name="Faktorial RAKL") %>%
  set_units(kelompok=3,
            petak=nested_in(kelompok, 12)) %>%
  set_trts(varietas=vari,
```

```

      pupuk=pupu) %>%
allot_trts(c(varietas,pupuk)~petak) %>%
assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desFaktRAKL,n=10))

```

kelompok	petak	varietas	pupuk
kelompok1	petak1	V3	N0
kelompok1	petak2	V1	N1
kelompok1	petak3	V3	N3
kelompok1	petak4	V2	N0
kelompok1	petak5	V1	N0
kelompok1	petak6	V1	N2
kelompok1	petak7	V2	N2
kelompok1	petak8	V2	N3
kelompok1	petak9	V1	N3
kelompok1	petak10	V2	N1

Plot dari rancangan tersebut dapat dilihat:

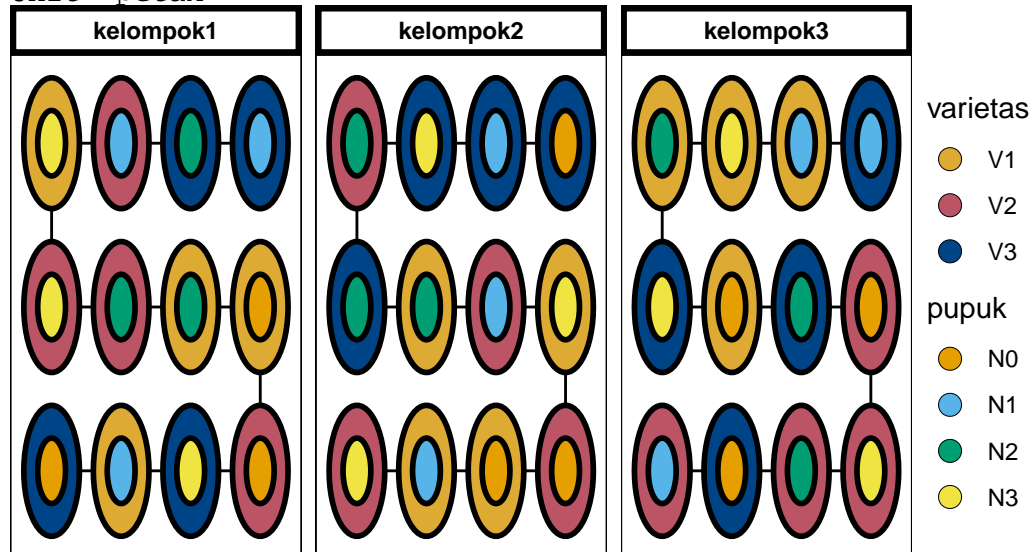
```

deggust::autoplot(desFaktRAKL)

```

Faktorial RAKL

Unit: petak



5.2.2 Model Linear Aditif

Model bagi faktorial RAL adalah sebagai berikut:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \rho_k + \varepsilon_{ijk}$$

Dengan:

1. Y_{ijk} adalah nilai pengamatan pada faktor A taraf ke-i dan faktor B taraf ke-j pada ulangan ke-k.
2. μ adalah komponen aditif dari rata-rata umum.
3. α_i adalah komponen aditif dari pengaruh faktor A pada taraf ke-i.
4. β_j adalah komponen aditif dari pengaruh faktor B pada taraf ke-j.
5. $(\alpha\beta)_{ij}$ adalah komponen interaksi faktor A dan faktor B.
6. ρ_k adalah pengaruh aditif dari kelompok dan diasumsikan tidak berinteraksi dengan perlakuan (bersifat aditif) / pengaruh kelompok ke-k
7. $\varepsilon_{ijk} \sim N(0, \sigma_\varepsilon^2)$ adalah komponen acak pada tiap pengamatan

Oleh karena itu, terdapat empat hipotesis yang diuji, yaitu hipotesis mengenai pengaruh faktor A, pengaruh faktor B, pengaruh interaksi, dan pengaruh kelompok:

Pengaruh faktor A:

$$H_0 : \alpha_1 = \dots = \alpha_a = 0$$

(faktor A tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada satu } i \text{ di mana } \alpha_i \neq 0$$

Pengaruh faktor B:

$$H_0 : \beta_1 = \dots = \beta_b = 0$$

(faktor B tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada satu } j \text{ di mana } \beta_j \neq 0$$

Pengaruh interaksi:

$$H_0 : (\alpha\beta)_{11} = (\alpha\beta)_{12} = \dots = (\alpha\beta)_{ab} = 0$$

(Interaksi faktor A dan B tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada sepasang } (i, j) \text{ di mana } (\alpha\beta)_{ij} \neq 0$$

Pengaruh pengelompokan:

$$H_0 : \rho_1 = \dots = \rho_r = 0$$

(blok tidak berpengaruh pada respon yang diamati)

$$H_1 : \text{paling sedikit ada satu } k \text{ di mana } \rho_k \neq 0$$

Hipotesis tersebut diuji dengan uji-F, yang disimpulkan di tabel sidik ragam:

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
A	a-1	JKA	JKP/dbA	KTA/KTG	
B	b-1	JKB	JKB/dbB	KTB/KTG	
A*B	(a-1)(b-1)	JKAB	JKAB/dbAB	KTAB/KTG	
Blok	(r-1)	JKK	JKK/dbK		
Galat	(ab-1)(r-1)	JKG	JKG/dbG		
Total	abr-1	JKT			

Jumlah kuadrat tersebut dihitung dengan:

$$\begin{aligned}
FK &= \frac{y_{...}^2}{abr} \\
JKT &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (y_{ijk} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r y_{ijk}^2 - FK \\
JKA &= \sum_{i=1}^a (\bar{y}_{i..} - \bar{y}_{...})^2 = \sum_{i=1}^a \frac{y_{i..}^2}{br} - FK \\
JKB &= \sum_{j=1}^b (\bar{y}_{.j.} - \bar{y}_{...})^2 = \sum_{j=1}^b \frac{y_{.j.}^2}{ar} - FK \\
JKP &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \frac{y_{ij.}^2}{r} - FK \\
JKAB &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 - JKA - JKB \\
&= JKP - JKB - JKA \\
JKK &= \sum_{k=1}^r (\bar{y}_{...k} - \bar{y}_{...})^2 = \sum_{k=1}^r \frac{y_{...k}^2}{ab} - FK \\
JKG &= JKP - JKK - JKT
\end{aligned}$$

5.2.3 ANOVA dengan R

Seorang peneliti akan melakukan percobaan pada sebuah tanaman dengan mengamati pertumbuhan tinggi tanaman di sebuah green house. Tanaman diberikan 2 jenis perlakuan yang pertama adalah jenis varietas yang terdiri dari Varietas 1(V1), Varietas 2(V2), dan Varietas 3(V3) dan perlakuan kedua adalah dosis pupuk terdiri dari N1, N2, N3, N4, N5. Percobaan ini dilakukan di 3 spot berbeda. Rancangan apakah yang sebaiknya digunakan peneliti? Bagaimana alur percobaan jika dilakukan sesuai rancangan yang anda sarankan? Lalu uji apakah perlakuan yang dicobakan berpengaruh signifikan, jika didapatkan data seperti dibawah ini?

Karena percobaan dilakukan di tiga *spot* berebeda, maka rancangan adalah faktorial RAKL. Ambil data:

```
library(googleSheets4)
```

```
FactRAKL<-read_sheet("https://docs.google.com/spreadsheets/d/1Pq0sMGjb6aBwF-3NswxsXk3-eevU
knitr::kable(FactRAKL)
```

Varietas	kelompok	N1	N2	N3	N4	N5
V1	1	0.9	1.2	1.3	1.8	1.1
V1	2	0.9	1.3	1.5	1.9	1.4
V1	3	1.0	1.2	1.4	2.1	1.2
V2	1	0.9	1.1	1.3	1.6	1.9
V2	2	0.8	0.9	1.5	1.3	1.6
V2	3	0.8	0.9	1.1	1.1	1.5
V3	1	0.9	1.4	1.3	1.4	1.2
V3	2	1.0	1.2	1.4	1.5	1.1
V3	3	0.7	1.0	1.4	1.4	1.3

Lalu, melt data frame tersebut. Seperti biasa, id.vars merupakan pembeda individu di format wide yang menjadi bentuk awal data frame (dalam kasus ini, tiap individu adalah kombinasi Varietas dan Kelompok). Lalu, value.vars merupakan kolom-kolom yang digabung menjadi satu, dalam kasus ini pupuk dari N1 sampai N5.

```
library(reshape2)

FactRAKL2<-melt(FactRAKL,
  #variabel yang membedakan tiap baris di tabulasi asli:
  id.vars=c("Varietas","kelompok"),
  value.vars=c("N1","N2","N3","N4","N5"),
  value.name="Tinggi")
colnames(FactRAKL2)[3]<-"Pupuk" #kolom ketiga akan disebut "variable"
knitr::kable(FactRAKL2)
```

Varietas	kelompok	Pupuk	Tinggi
V1	1	N1	0.9
V1	2	N1	0.9
V1	3	N1	1.0
V2	1	N1	0.9
V2	2	N1	0.8
V2	3	N1	0.8
V3	1	N1	0.9
V3	2	N1	1.0
V3	3	N1	0.7
V1	1	N2	1.2
V1	2	N2	1.3
V1	3	N2	1.2
V2	1	N2	1.1

Varietas	kelompok	Pupuk	Tinggi
V2	2	N2	0.9
V2	3	N2	0.9
V3	1	N2	1.4
V3	2	N2	1.2
V3	3	N2	1.0
V1	1	N3	1.3
V1	2	N3	1.5
V1	3	N3	1.4
V2	1	N3	1.3
V2	2	N3	1.5
V2	3	N3	1.1
V3	1	N3	1.3
V3	2	N3	1.4
V3	3	N3	1.4
V1	1	N4	1.8
V1	2	N4	1.9
V1	3	N4	2.1
V2	1	N4	1.6
V2	2	N4	1.3
V2	3	N4	1.1
V3	1	N4	1.4
V3	2	N4	1.5
V3	3	N4	1.4
V1	1	N5	1.1
V1	2	N5	1.4
V1	3	N5	1.2
V2	1	N5	1.9
V2	2	N5	1.6
V2	3	N5	1.5
V3	1	N5	1.2
V3	2	N5	1.1
V3	3	N5	1.3

Setelah tabulasi data benar, pastikan tiap faktor sudah dibuat menjadi objek data yang benar; lakukan ANOVA dan lihat hasil:

```
FactRAKL2$Varietas<-as.factor(FactRAKL2$Varietas)
FactRAKL2$Pupuk<-as.factor(FactRAKL2$Pupuk)
FactRAKL2$kelompok<-as.factor(FactRAKL2$kelompok)
aovFactRAKL<-aov(Tinggi~Varietas*Pupuk+kelompok,FactRAKL2)
```

```
summary(aovFactRAKL)
```

```

              Df Sum Sq Mean Sq F value    Pr(>F)
Varietas      2  0.1693   0.0847    4.316   0.0232 *
Pupuk         4  2.4902   0.6226   31.732 4.95e-10 ***
kelompok     2  0.0640   0.0320    1.631   0.2138
Varietas:Pupuk 8  1.0151   0.1269    6.468 8.98e-05 ***
Residuals    28  0.5493   0.0196
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Operasi-operasi lanjutan, seperti uji asumsi, uji lanjut, ataupun pembuatan plot interaksi, dapat diadopsi dari contoh-contoh sebelumnya.

```

SumTabRAKFakt<-reshape2::dcast(aggregate(Tinggi~Varietas+Pupuk,data=FactRAKL2,FUN=mean),Va
knitr::kable(SumTabRAKFakt)

```

Varietas	N1	N2	N3	N4	N5
V1	0.9333333	1.2333333	1.400000	1.933333	1.233333
V2	0.8333333	0.9666667	1.300000	1.333333	1.666667
V3	0.8666667	1.2000000	1.366667	1.433333	1.200000

Dan plot interaksinya:

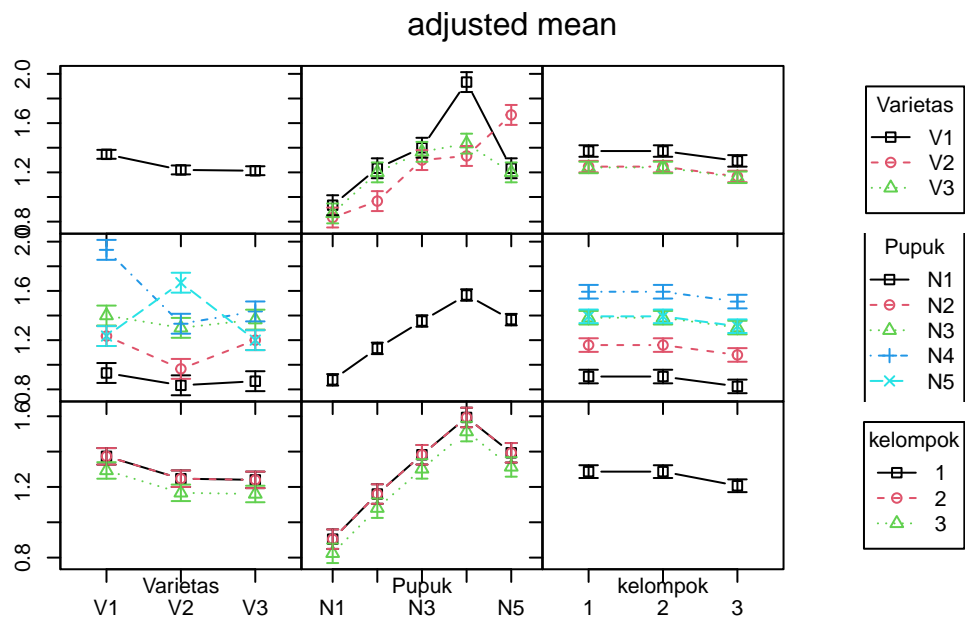
```

library(phia)
mod2<-lm(Tinggi~Varietas*Pupuk+kelompok,data=FactRAKL2)
im2=interactionMeans(mod2)
knitr::kable(head(im2),n=5) # tabel

```

Varietas	Pupuk	kelompok	adjusted mean	std. error
V1	N1	1	0.9600000	0.0860909
V2	N1	1	0.8600000	0.0860909
V3	N1	1	0.8933333	0.0860909
V1	N2	1	1.2600000	0.0860909
V2	N2	1	0.9933333	0.0860909
V3	N2	1	1.2266667	0.0860909

```
plot(im2) # plot
```



Terlihat bahwa pengelompokan memenuhi asumsi tak ada interaksi.

6 Rancangan Petak Terpisah (Split Plot Design)

Dalam percobaan dua faktor, kombinasi perlakuan belum tentu diacak secara sempurna terhadap unit percobaan. Pengacakan sempurna berarti semua kombinasi perlakuan dibuat dan diacak pada semua unit percobaan (atau pada suatu blok di Split Plot RAKL). Dalam suatu rancangan split plot, pengacakan dilakukan **bertahap**. **Petak utama** diacak terlebih dahulu, lalu faktor yang ditempatkan sebagai **anak petak** diacak di tiap petak utama. Rancangan ini dipilih karena beberapa alasan:

1. Tingkatan kepentingan dari faktor yang dilibatkan. Di Tabel ANOVA, akan terlihat bahwa derajat bebas galat petak utama lebih kecil dari anak petak. Ini berarti faktor pembagi jumlah kuadrat galat akan lebih kecil di petak utama, sehingga kuadrat tengah galat bisa saja lebih besar. Berarti, keragaman (atau kesalahan yang muncul dari faktor acak) lebih besar di petak utama sehingga ketepatan uji di petak utama lebih rendah. Ini dapat terjadi jika peneliti lebih tertarik pada suatu efek (misal, dengan efek varietas daripada lokasi). Oleh karena itu, efek varietas yang harus diduga dengan lebih tepat menjadi anak petak.
2. Nisbah pengaruh faktor. Jika satu faktor memiliki pengaruh besar, tidak perlu ketepatan tinggi untuk menangkap pengaruh tersebut. Misal diketahui bahan berpengaruh besar pada kualitas suatu pedang (misal, pedang baja lebih kuat dari pedang besi), dan pengaruh waktu tempa lebih kecil. Bahan dapat menjadi petak utama.
3. Pengembangan dari percobaan yang sudah berjalan. Misal ada percobaan yang mengacak dosen-dosen ke kelas tertentu, lalu di dalam kelas tersebut peneliti tertarik melihat pengaruh bimbingan belajar. Oleh karena itu, kelas yang mengandung dosen menjadi petak utama, lalu bimbingan belajar diacak ke siswa yang menjadi anak petak.
4. Kendala pengacakan di lapangan. Misal, salah satu faktor tidak bisa atau tidak efisien jika diacak secara karena taraf faktor membutuhkan unit yang besar. Misal, pengolahan lahan membutuhkan traktor, sehingga harus menjadi petak utama dan varietas menjadi anak petak.
5. Lokasi menjadi petak utama dan perlakuan menjadi anak petak. Misal dicobakan varietas tanaman di kampus IPB Dramaga dan di kampus IPB Baranangsiang.
6. Waktu (musim, tahun) menjadi petak utama, dan perlakuan menjadi anak petak. Semisal di diacak beberapa kegiatan besar untuk tiap semester, lalu di tiap semester berkegiatan besar tersebut diacak anggota panitianya.

7. Repeated measurements. Perlakuan menjadi petak utama, lalu waktu-waktu tertentu pengamatan menjadi anak petak

6.1 Split Plot RAL

6.1.1 Pengacakan

Pengacakan pada rancangan petak terpisah di R dapat langsung dilakukan menggunakan fungsi `design.split` dari library `agricolae`. Akan digunakan studi kasus berikut:

Dalam usaha menjaga kesehatan lingkungan dilakukan percobaan dengan menggunakan berbagai jenis tanaman untuk menyerap debu di udara. Jenis tanaman yang digunakan antara lain tanaman berdaun kecil licin (J1 dan J2), berdaun kecil kasar (J3, J4), berdaun lebar licin (J5, J6), berdaun lebar kasar (J7, J8), dan berdaun jarum (J9, J10). Percobaan ini dicobakan pada dua lokasi (Cipedak dan Gatot Subroto). Setiap perlakuan diulang 3 kali dan unit-unit percobaan yang digunakan diasumsikan homogen. Dari percobaan ini ingin diketahui jenis tanaman yang mampu menyerap debu paling efektif dan di lokasi mana? Peubah respons yang dicatat dari percobaan ini adalah konsentrasi debu (ppm) yang melekat pada daun tanaman.

Untuk melakukan pengacakan, bangkitkan taraf kedua faktor terlebih dahulu:

```
Lokasi<-c("Cipedak","Gatot S.")
Jenis<-paste("J",seq(10),sep="")
```

Pengacakan secara cepat dapat dilakukan dengan menggunakan `design.split` dari library `agricolae`. Pastikan design berupa CRD (completely randomized design) jika menggunakan split plot RAL. Argumen fungsi diurutkan sebagai berikut karena lokasi merupakan petak utama, jenis anak petak, dan ada tiga ulangan.

```
library(agricolae)

bagan<-design.split(Lokasi, Jenis,r=3, design="crd",serie = 0,
  seed = 0, kinds = "Super-Duper", first=TRUE,randomization=TRUE)$book #(utama,
knitr::kable(head(bagan),n=10)
```

plots	splots	r	Lokasi	Jenis
1	1	1	Gatot S.	J2
1	2	1	Gatot S.	J10
1	3	1	Gatot S.	J8

plots	splots	r	Lokasi	Jenis
1	4	1	Gatot S.	J7
1	5	1	Gatot S.	J5
1	6	1	Gatot S.	J1

```
nrow(bagan)
```

```
[1] 60
```

Bagan memiliki 60 baris, yang masuk akal karena $A \times B \times r = 2 \times 10 \times 3 = 60$, di mana faktor A adalah lokasi, B adalah jenis, dan r ulangan. Jika ingin lebih detail, dapat diperhatikan bahwa pengacakan split plot dapat dilakukan sebagai berikut:

1. Pengacakan RAL untuk plot utama, dengan unit percobaan berjumlah $A \times r$.
2. RAKL dengan $A \times r$ blok, dengan B unit percobaan di blok tersebut.

Implementasi di agricolae sebagai berikut:

```
bagan1<-design.crd(Lokasi,r=3,serie=0)
knitr::kable(bagan1$book)
```

plots	r	Lokasi
1	1	Cipedak
2	1	Gatot S.
3	2	Cipedak
4	3	Cipedak
5	2	Gatot S.
6	3	Gatot S.

Lalu, fase kedua dari pengacakan. Ada $2 \times 3 = 6$ blok:

```
bagan2<-design.rcbd(Jenis,r=6)
knitr::kable(head(bagan2$book),n=10)
```

plots	block	Jenis
101	1	J2
102	1	J6

plots	block	Jenis
103	1	J1
104	1	J3
105	1	J7
106	1	J10

```
nrow(bagan2$book)
```

```
[1] 60
```

Lakukan merge. Note bahwa tiap plot di pengacakan pertama akan menjadi blok, sehingga merge di x menggunakan plot dan di y menggunakan blok:

```
baganfin<-merge(bagan1$book,bagan2$book,by.x="plots",by.y="block")
knitr::kable(head(baganfin),n=10)
```

plots	r	Lokasi	plots.y	Jenis
1	1	Cipedak	101	J2
1	1	Cipedak	102	J6
1	1	Cipedak	103	J1
1	1	Cipedak	104	J3
1	1	Cipedak	105	J7
1	1	Cipedak	106	J10

```
nrow(baganfin)
```

```
[1] 60
```

Sehingga terbuat bagan final. Bagan tersebut dapat dilihat dengan beberapa format lain, seperti:

```
bag2<-bagan[,-c(1,2)]
knitr::kables(list(
  knitr::kable(reshape2::dcast(bag2,Lokasi~Jenis)), #lokasi vs jenis (berapa jenis tertent
  knitr::kable(reshape2::dcast(bag2,Lokasi+r~Jenis)) #lokasi, ulangan vs jenis (bagan, tap
),
```

Table 6.5: Dua cara menunjukkan bagan percobaan

Lokasi	J1	J10	J2	J3	J4	J5	J6	J7	J8	J9
Cipedak	3	3	3	3	3	3	3	3	3	3
Gatot S.	3	3	3	3	3	3	3	3	3	3

Lokasi	r	J1	J10	J2	J3	J4	J5	J6
Cipedak	1	J1	J10	J2	J3	J4	J5	J6
Cipedak	2	J1	J10	J2	J3	J4	J5	J6
Cipedak	3	J1	J10	J2	J3	J4	J5	J6
Gatot S.	1	J1	J10	J2	J3	J4	J5	J6
Gatot S.	2	J1	J10	J2	J3	J4	J5	J6
Gatot S.	3	J1	J10	J2	J3	J4	J5	J6

```
caption="Dua cara menunjukkan bagan percobaan"
)
```

Hal tersebut dapat juga dilakukan dengan alternatif pengacakan dua fase:

```
knitr::kable(cbind(bagan1$book,bagan2$sketch))
```

plots	r	Lokasi	1	2	3	4	5	6	7	8	9	10
1	1	Cipedak	J2	J6	J1	J3	J7	J10	J9	J4	J5	J8
2	1	Gatot S.	J6	J5	J4	J3	J9	J10	J7	J1	J8	J2
3	2	Cipedak	J10	J8	J2	J4	J9	J5	J3	J1	J7	J6
4	3	Cipedak	J9	J10	J8	J5	J2	J6	J4	J7	J1	J3
5	2	Gatot S.	J9	J10	J4	J8	J3	J6	J1	J2	J5	J7
6	3	Gatot S.	J5	J7	J9	J10	J8	J4	J6	J1	J3	J2

6.1.1.1 Pengacakan edibble

Untuk split plot RAL, dapat dibuat dua unit di `set_units()`. Pertama adalah petak utama, dan anak petak berupa unit yang `nested_in()` petak utama. Taraf lokasi diacak ke petak utama (`lokasi~mainplot`), dan jenis ke anak petak (`jenis~unit`):

```
library(edibble)

desSPRAL<-design(name="Lokasi dan Jenis vs. Penyerapan Debu") %>%
  set_units(mainplot=2,
            unit=nested_in(mainplot, 10)) %>%
  set_trts(lokasi=Lokasi,
            jenis=Jenis) %>%
  allot_trts(lokasi~mainplot,
```

```

      jenis~unit) %>%
assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desSPRAL,n=10))

```

mainplot	unit	lokasi	jenis
mainplot1	unit1	Gatot S.	J6
mainplot1	unit2	Gatot S.	J8
mainplot1	unit3	Gatot S.	J1
mainplot1	unit4	Gatot S.	J10
mainplot1	unit5	Gatot S.	J3
mainplot1	unit6	Gatot S.	J7
mainplot1	unit7	Gatot S.	J9
mainplot1	unit8	Gatot S.	J2
mainplot1	unit9	Gatot S.	J4
mainplot1	unit10	Gatot S.	J5

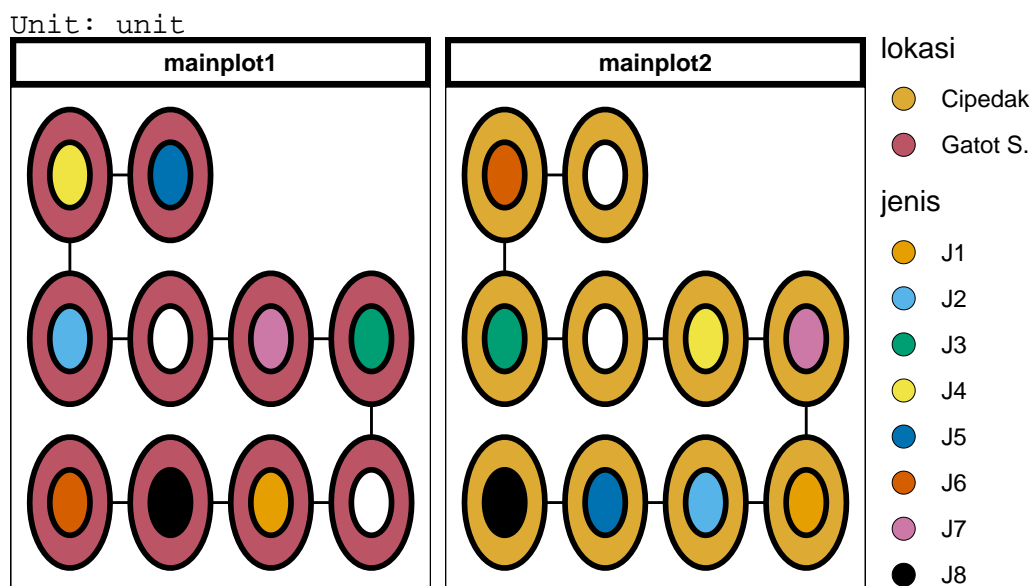
Plot dari rancangan tersebut dapat dilihat:

```

deggust::autoplot(desSPRAL)

```

Lokasi dan Jenis vs. Penyerapan Debu



Edible juga dapat membuat rancangan split plot secara *default*:

```
spd<-takeout(menu_split(t1 = 2, t2 = 10, r = 1, seed=420))
examine_recipe(spd)
```

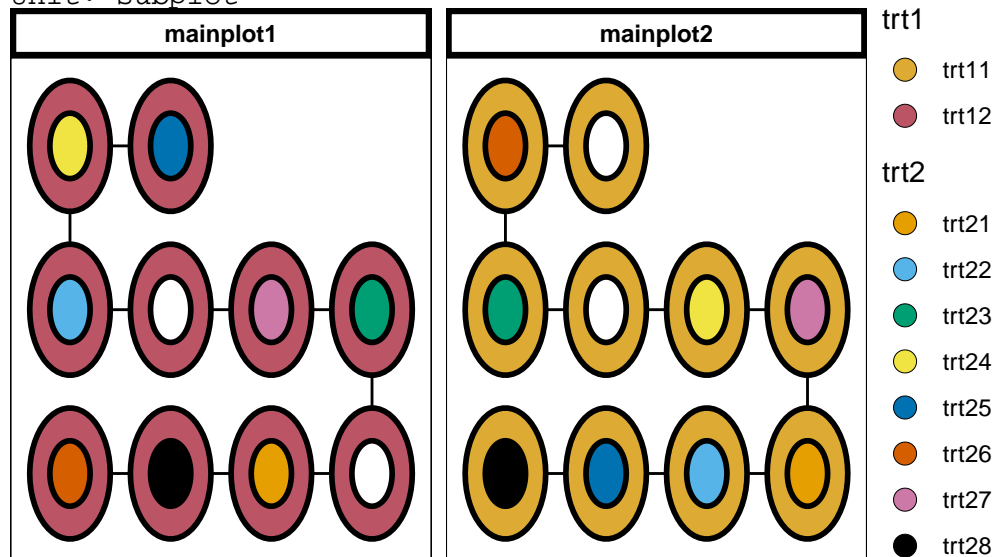
```
design("Split-Plot Design | Split-Unit Design") %>%
  set_units(mainplot = 2,
            subplot = nested_in(mainplot, 10)) %>%
  set_trts(trt1 = 2,
            trt2 = 10) %>%
  allot_trts(trt1 ~ mainplot,
             trt2 ~ subplot) %>%
  assign_trts("random", seed = 420) %>%
  serve_table()
```

Dapat dilihat bahwa cara pembuatan rancangan sama dengan yang dibuat sebelumnya. Plot juga sama:

```
degust::autoplot(spd)
```

Split-Plot Design | Split-Unit Design

Unit: subplot



6.1.2 ANOVA

Model linear aditif bagi split plot RAL adalah:

$$y_{ijk} = \mu + \alpha_i + \delta_{k(i)} + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Dengan:

1. y_{ijk} adalah nilai pengamatan pada faktor A taraf ke-i dan faktor B taraf ke-j pada ulangan ke-k.
2. μ komponen aditif dari rata-rata umum.
3. α_i pengaruh utama faktor A. Diasumsikan $\sum_{i=1}^a \alpha_i = 0$ bagi model tetap dan $\alpha \sim N(0, \sigma_a^2)$ bagi model acak.
4. β_j pengaruh utama faktor B. Diasumsikan $\sum_{j=1}^b \beta_j = 0$ bagi model tetap dan $\beta \sim N(0, \sigma_b^2)$ bagi model acak.
5. $(\alpha\beta)_{ij}$ komponen interaksi faktor A dan faktor B. Diasumsikan $\sum_{i=1}^p \sum_{j=1}^t (\alpha\beta)_{ij} = 0$ bagi model tetap dan $(\alpha\beta)_{ij} \sim N(0, \sigma_{ab}^2)$ bagi model acak.
6. $\delta_{k(i)} \sim N(0, \sigma_\delta^2)$ komponen acak dari petak utama.
7. $\epsilon_{ijk} \sim N(0, \sigma^2)$ komponen acak anak petak.

Dan hipotesis pengaruh petak utama:

$$\begin{aligned} H_0 : \alpha_1 = \dots = \alpha_a = 0 \\ \text{faktor A tidak berpengaruh pada respon} \\ H_1 : \text{Paling sedikit ada satu } i \text{ di mana } \alpha_i \neq 0 \end{aligned}$$

Anak petak:

$$\begin{aligned} H_0 : \beta_1 = \dots = \beta_b = 0 \\ \text{faktor B tidak berpengaruh pada respon} \\ H_1 : \text{Paling sedikit ada satu } j \text{ di mana } \beta_j \neq 0 \end{aligned}$$

Dan interaksi:

$$\begin{aligned} H_0 : (\alpha\beta)_{11} = (\alpha\beta)_{12} = \dots = (\alpha\beta)_{ab} = 0 \\ \text{interaksi tidak berpengaruh pada respon} \\ H_1 : \text{Paling sedikit ada sepasang } ij \text{ di mana } (\alpha\beta)_{ij} \neq 0 \end{aligned}$$

Dan tabel ANOVA:

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
A	a-1	JKA	JKA/dbA	KTGA/KTGA	
Galat (a)	a(r-1)	JKGA	JKGA/dbGA		
B	(b-1)	JKB	JKB/dbB	KTGB/KTGB	
AB	(a-1)(b-1)	JKAB	JKAB/dbAB	KTAB/KTGB	
Galat (b)	a(b-1)(r-1)	JKGB	JKGB/dbGB		
Total	abr-1	JKT			

Jumlah kuadrat tersebut dihitung dengan:

$$\begin{aligned}
FK &= \frac{y_{...}^2}{abr} \\
JKT &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (y_{ijk} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r y_{ijk}^2 - FK \\
JKST &= \sum_{i=1}^a \sum_{k=1}^r (\bar{y}_{i.K} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{k=1}^r \frac{y_{i.k}^2}{b} - FK \\
JKA &= \sum_{i=1}^a (\bar{y}_{i..} - \bar{y}_{...})^2 = \sum_{i=1}^a \frac{y_{i..}^2}{br} - FK \\
JKG_a &= JKST - JKA \\
JKB &= \sum_{j=1}^b (\bar{y}_{.j.} - \bar{y}_{...})^2 = \sum_{j=1}^b \frac{y_{.j.}^2}{ar} - FK \\
JKP &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \frac{y_{ij.}^2}{r} - FK \\
JKAB &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 - JKA - JKB \\
&= JKP - JKB - JKA \\
JKG_b &= JKT - JKP - JKG_a
\end{aligned}$$

6.1.3 Analisis menggunakan R

Akses data:

```
library(googleheets4)
SPlot<-read_sheet("https://docs.google.com/spreadsheets/d/1CL0DZ8Ub_sTvDcj-jy_UR3P3KKY1vpt")
```

! Using an auto-discovered, cached token.

To suppress this message, modify your code or options to clearly consent to the use of a cached token.

See gargle's "Non-interactive auth" vignette for more details:

<<https://gargle.r-lib.org/articles/non-interactive-auth.html>>

i The googlesheets4 package is using a cached token for 'muh.ammarsahab@apps.ipb.ac.id'.

v Reading from "SplitPlots".

v Range 'Sheet1'.

```
knitr::kable(head(SPlot),n=10)
```

No	Lokasi	Jenis	Ulangan	Konsentrasi debu
1	Cipedak	J1	1	0.0312
2	Cipedak	J1	2	0.0317
3	Cipedak	J1	3	0.0321
4	Cipedak	J2	1	0.0270
5	Cipedak	J2	2	0.0272
6	Cipedak	J2	3	0.0277

Akan dilakukan dua hal untuk membersihkan data. Pertama, satuan respon akan dikalikan 1000 karena satuan awalnya relatif kecil. Jika dikalikan 1000 jumlah kuadrat dan kuadrat tengah akan terlihat lebih jelas:

```
SPlot$`Konsentrasi debu`<-SPlot$`Konsentrasi debu`*1000
```

Lalu, pastikan semua faktor telah menjadi... faktor:

```
SPlot$Ulangan<-as.factor(SPlot$Ulangan)
SPlot$Jenis<-as.factor(SPlot$Jenis)
SPlot$Lokasi<-as.factor(SPlot$Lokasi)
```


Lakukan ANOVA. Seperti di percobaan faktorial, sintaks $a * b$ akan menghasilkan pengaruh faktor dan interaksi secara bersamaan. Oleh karena itu, tulis saja sintaks tersebut tanpa harus menspesifikasi pengaruh tiap faktor. Terdapat sintaks yang berbeda, yaitu di Error(Ulangan:Lokasi). Pada dasarnya sintaks ini memerintahkan R untuk menghitung ANOVA bagi plot utama. Dihitung JKG_{Lokasi} dan pengaruh lokasi diuji menggunakan kuantitas tersebut. Tabel ANOVA akan terbagi dua menjadi Error: Ulangan:Lokasi - di plot utama, dan Error: Within - di dalam plot utama, dalam kata lain di anak petak. Pastikan ulangan sudah merupakan faktor (jika belum, derajat bebas akan beda).

```
aovSPlot1<-aov(`Konsentrasi debu`~Lokasi*Jenis+Error(Ulangan:Lokasi),data=SPlot)
```

```
Warning in aov(`Konsentrasi debu` ~ Lokasi * Jenis + Error(Ulangan:Lokasi), :
Error() model is singular
```

```
summary(aovSPlot1)
```

```
Error: Ulangan:Lokasi
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Lokasi	1	105404	105404	45587	2.89e-09 ***
Residuals	4	9	2		

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Error: Within
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Jenis	9	160762	17862	343631	<2e-16 ***
Lokasi:Jenis	9	20732	2304	44314	<2e-16 ***
Residuals	36	2	0		

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Terbentuk tabel ANOVA yang mirip dengan Excel, dengan kesimpulan sama yaitu tolak H_0 bagi tiap pengaruh. Ini berarti lokasi, jenis, dan interaksi lokasi dengan jenis berpengaruh pada respon. ANOVA Split Plot RAL dapat juga dianalisis melalui package `lme4`:

```
library(lme4)
```

```
Loading required package: Matrix
```

```
aovSPlot2<-lmer(`Konsentrasi debu`~Lokasi*Jenis+(1|Ulangan:Lokasi),data=SPlot)
knitr::kable(anova(aovSPlot2))
```

	npar	Sum Sq	Mean Sq	F value
Lokasi	1	2369.655	2369.655	45586.53
Jenis	9	160762.103	17862.456	343631.14
Lokasi:Jenis	9	20731.646	2303.516	44314.17

Nilai F dari ANOVA tersebut sama dengan sebelumnya. Karena setidaknya salah satu pengaruh jenis dan lokasi beda dari nol, dapat dibuat kontras sebagai uji lanjut:

```
contrasts(SPlot$Jenis)<-cbind(c(1, -4, 1, 1, 1, 1, 1, 1, 1, -4), c(0, 1, 0, 0, 0, 0, 0, 0, 0, 0),
                             c(1, 0, 1, 1, 1, -1, -1, -1, -1, 0), c(1, 0, 1, -1, -1, 0, 0, 0, 0, 0),
                             c(1, 0, -1, 0, 0, 0, 0, 0, 0, 0), c(0, 0, 0, 1, -1, 0, 0, 0, 0, 0),
                             c(0, 0, 0, 0, 0, 1, 1, -1, -1, 0), c(0, 0, 0, 0, 0, 1, -1, 0, 0, 0),
                             c(0, 0, 0, 0, 0, 0, 0, 1, -1, 0))
summary(aovSPlot1,split=list(Jenis=list("Jarum vs Lainnya"=1, "J9 vs J10"=2, "Kecil vs Lebar"=3,
"J1 vs J2 (kecil licin)"=5, "J3 vs J4 (kecil kasar)"=6, "licin vs kasar (lebar)"=7,
"J5 vs J6 (lebar licin)"=8, "J7 vs J8 (lebar kasar)"=9)))
```

Error: Ulangan:Lokasi

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Lokasi	1	105404	105404	45587	2.89e-09 ***
Residuals	4	9	2		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within

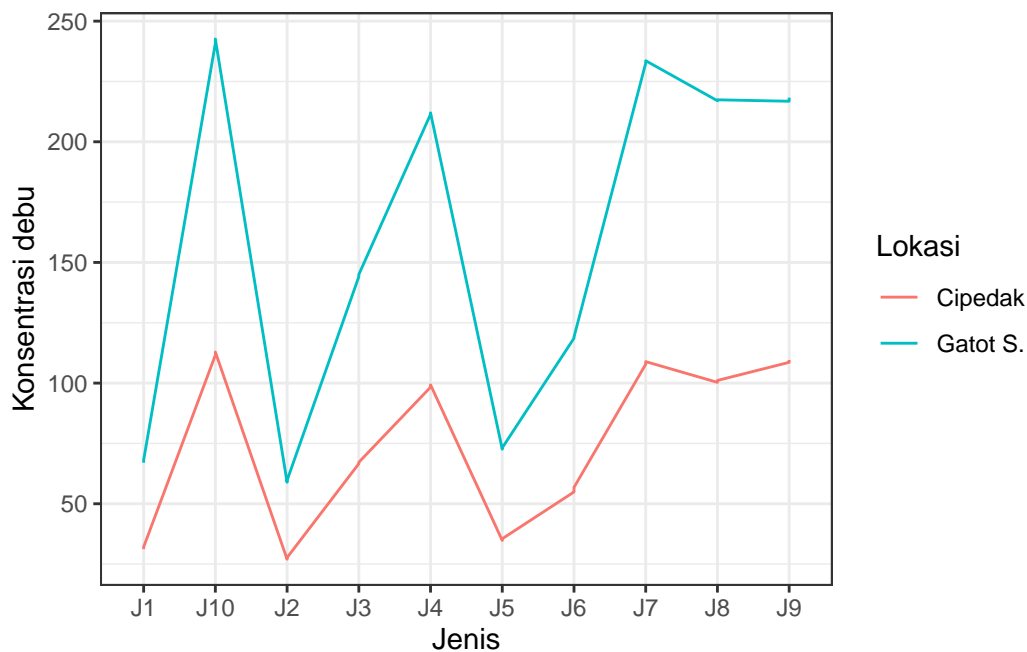
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Jenis	9	160762	17862	343631	<2e-16 ***
Jenis: Jarum vs Lainnya	1	24359	24359	468603	<2e-16 ***
Jenis: J9 vs J10	1	29839	29839	574023	<2e-16 ***
Jenis: Kecil vs Lebar	1	1048	1048	20168	<2e-16 ***
Jenis: Licin vs Kasar (kecil)	1	8740	8740	168142	<2e-16 ***
Jenis: J1 vs J2 (kecil licin)	1	25855	25855	497393	<2e-16 ***
Jenis: J3 vs J4 (kecil kasar)	1	11248	11248	216388	<2e-16 ***
Jenis: licin vs kasar (lebar)	1	9842	9842	189338	<2e-16 ***
Jenis: J5 vs J6 (lebar licin)	1	11025	11025	212095	<2e-16 ***
Jenis: J7 vs J8 (lebar kasar)	1	38806	38806	746532	<2e-16 ***

Lokasi:Jenis	9	20732	2304	44314	<2e-16	***
Lokasi:Jenis: Jarum vs Lainnya	1	3456	3456	66495	<2e-16	***
Lokasi:Jenis: J9 vs J10	1	3680	3680	70799	<2e-16	***
Lokasi:Jenis: Kecil vs Lebar	1	84	84	1618	<2e-16	***
Lokasi:Jenis: Licin vs Kasar (kecil)	1	1301	1301	25021	<2e-16	***
Lokasi:Jenis: J1 vs J2 (kecil licin)	1	3374	3374	64907	<2e-16	***
Lokasi:Jenis: J3 vs J4 (kecil kasar)	1	1325	1325	25496	<2e-16	***
Lokasi:Jenis: licin vs kasar (lebar)	1	1606	1606	30887	<2e-16	***
Lokasi:Jenis: J5 vs J6 (lebar licin)	1	1960	1960	37697	<2e-16	***
Lokasi:Jenis: J7 vs J8 (lebar kasar)	1	3946	3946	75908	<2e-16	***
Residuals	36	2	0			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Semua kontras menolak H_0 . Ini berarti jenis berbeda yang hendak dibandingkan benar-benar beda dalam hal konsentrasi debu. Lalu, dapat dibuat plot interaksi:

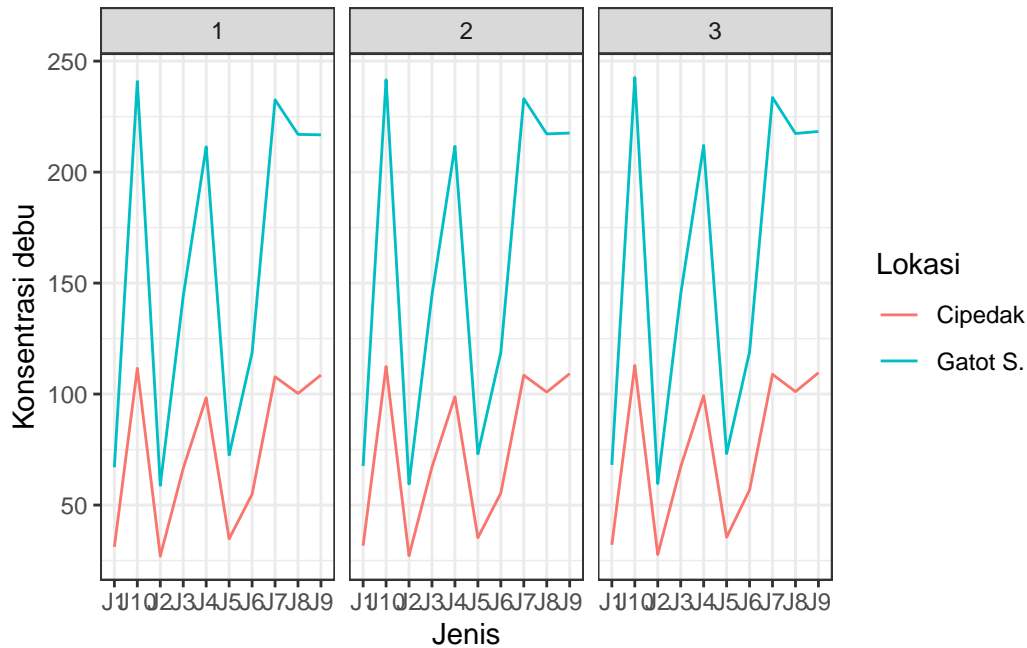
```
library(ggplot2)
ggplot2::ggplot(aes(x = Jenis, y = `Konsentrasi debu`, group = Lokasi, colour = Lokasi), data = data)
```



Dapat dilihat bahwa garis tak sepenuhnya sejajar walaupun tak pernah berpotongan. Misal, ada kenaikan konsentrasi debu yang lebih besar di Gatot Subroto daripada Cipedak antara

jenis tanaman yang sama (misal J5 vs. J6). Untuk memastikan bahwa RAL tepat, dapat dibuat facet per ulangan yang memisahkan plot menjadi per ulangan:

```
ggplot2::ggplot(aes(x = Jenis, y = `Konsentrasi debu`, group = Lokasi, colour = Lokasi), data = data) +
  facet_wrap(~ Ulangan) + theme_bw()
```



Terlihat bahwa pola interaksi relatif sama antara ulangan, sehingga dapat dianggap bahwa unit percobaan benar-benar homogen.

6.2 Split Plot RAK

Kondisi unit percobaan split plot belum tentu homogen. Oleh karena itu, dapat digunakan Split Plot RAK untuk mengatasi pengaruh keragaman yang muncul dari satu arah. Pengelompokan menjadi blok dilakukan di petak utama, dengan pengacakan anak petak relatif sama. Dalam pembahasan Split Plot RAK, akan digunakan kasus oats:

Produksi oat dari sebuah percobaan split-plot menggunakan tiga varietas dan empat taraf penggunaan nitrogen. Percobaan dilaksanakan di enam blok yang berisi tiga petak utama yang dibagi menjadi empat anak petak. Varietas diaplikasikan di petak utama dan pemupukan nitrogen diaplikasikan ke anak petak.

Peubah-peubah dalam dataset oats adalah:

1. B: Blok, dengan taraf I, II, III, IV, V dan VI.
2. V: Varietas, 3 taraf.
3. N: Perlakuan pupuk nitrogen. Taraf 0.0cwt, 0.2cwt, 0.4cwt, dan 0.6cwt.
4. Y: Produksi di 1/4 pound per sub-plot. Tiap sub-plot sebesar 1/80 acre.

Data tersebut ada dalam package MASS:

```
data(oats, package = "MASS")
str(oats)
```

```
'data.frame':  72 obs. of  4 variables:
 $ B: Factor w/ 6 levels "I","II","III",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ V: Factor w/ 3 levels "Golden.rain",...: 3 3 3 3 1 1 1 1 2 2 ...
 $ N: Factor w/ 4 levels "0.0cwt","0.2cwt",...: 1 2 3 4 1 2 3 4 1 2 ...
 $ Y: int  111 130 157 174 117 114 161 141 105 140 ...
```

6.2.1 Pengacakan

Pertama, ambil taraf-taraf dari varietas dan konsentrasi pupuk nitrogen:

```
V<-levels(oats$V)
N<-levels(oats$N)
```

Pengacakan secara cepat dapat dilakukan dengan menggunakan design.split dari library agricolae. Dalam kasus ini, design berupa RCBD (randomized complete block design) jika menggunakan split plot RAK. Argumen fungsi diurutkan sebagai berikut karena varietas merupakan petak utama, nitrogen anak petak, dan ada tiga blok.

```
library(agricolae)

bagan2<-design.split(V, N,r=6, design="rcbd",serie = 0,
                    seed = 0, kinds = "Super-Duper",
                    first=TRUE,randomization=TRUE)
#(utama, anak, ulangan)
knitr::kable(head(bagan2$book),n=10)
```

plots	splots	block	V	N
1	1	1	Golden.rain	0.2cwt
1	2	1	Golden.rain	0.0cwt

plots	splots	block	V	N
1	3	1	Golden.rain	0.4cwt
1	4	1	Golden.rain	0.6cwt
2	1	1	Marvellous	0.4cwt
2	2	1	Marvellous	0.2cwt

```
nrow(bagan2$book)
```

```
[1] 72
```

```
nrow(oats)
```

```
[1] 72
```

Jumlah baris di bagan sama dengan jumlah baris di data oats, yaitu $3 \times 4 \times 6 = 72$. Mirip dengan Split Plot RAL, pengacakan dapat dilakukan dengan algoritma:

1. Pengacakan RAKL untuk plot utama.
2. RAKL dengan $A \times r$ blok, dengan B unit percobaan di blok tersebut.

A adalah jumlah taraf faktor di plot utama dan B adalah jumlah taraf faktor di anak petak.

6.2.1.1 Pengacakan edibble

Untuk split plot RAKL, dapat dibuat tiga unit di `set_units()`. Pertama adalah blok, lalu petak utama yang `nested_in(blok)`, dan anak petak yang `nested_in()` petak utama. Taraf varietas diacak ke petak utama (`V~mainplot`), dan nitrogen ke anak petak (`jenis~subplot`):

```
library(edibble)

desSPRAKL<-design(name="Oats") %>%
  set_units(blok=6,
            mainplot=nested_in(blok,3),
            subplot=nested_in(mainplot, 4)) %>%
  set_trts(nitrogen=N,
           varietas=V) %>%
  allot_trts(varietas~mainplot,
```

```

nitrogen~subplot) %>%
assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desSPRAKL,n=10))

```

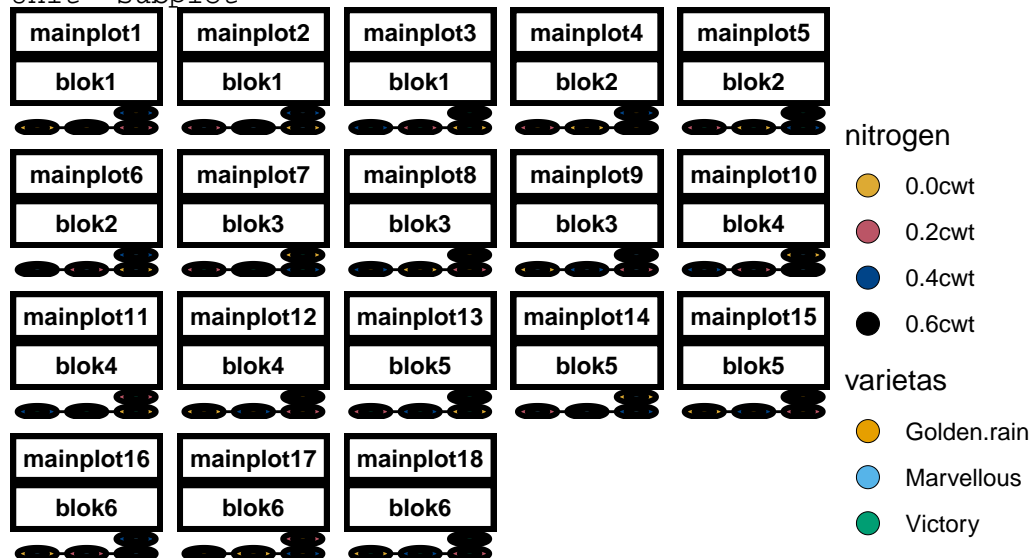
blok	mainplot	subplot	nitrogen	varietas
blok1	mainplot1	subplot1	0.0cwt	Golden.rain
blok1	mainplot1	subplot2	0.6cwt	Golden.rain
blok1	mainplot1	subplot3	0.2cwt	Golden.rain
blok1	mainplot1	subplot4	0.4cwt	Golden.rain
blok1	mainplot2	subplot5	0.2cwt	Marvellous
blok1	mainplot2	subplot6	0.6cwt	Marvellous
blok1	mainplot2	subplot7	0.0cwt	Marvellous
blok1	mainplot2	subplot8	0.4cwt	Marvellous
blok1	mainplot3	subplot9	0.4cwt	Victory
blok1	mainplot3	subplot10	0.2cwt	Victory

Plot dari rancangan tersebut dapat dilihat:

```
deggust::autoplot(desSPRAKL)
```

Oats

Unit: subplot

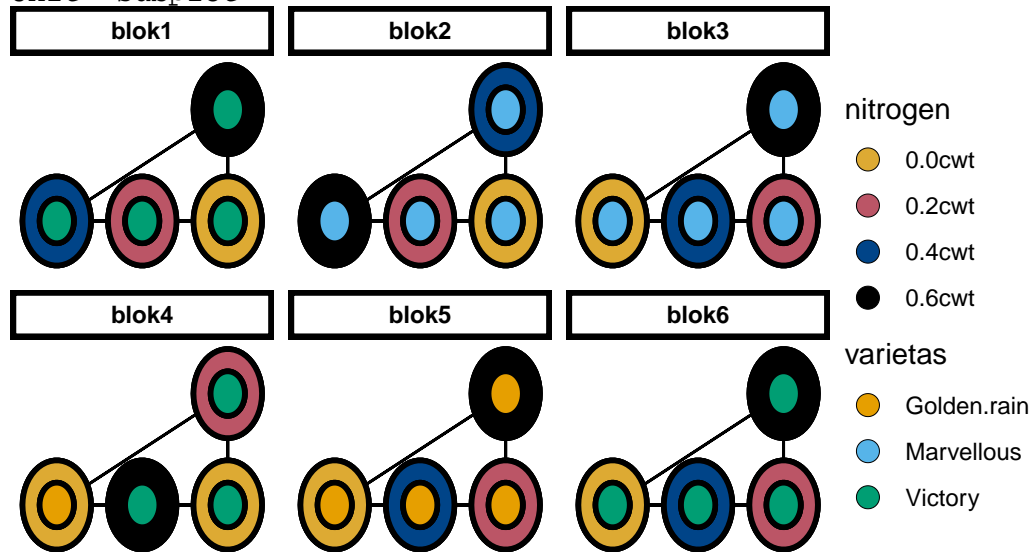


Agar plot tersebut terlihat lebih rapih, dapat digunakan `facet_wrap()` untuk memisahkan plot tersebut per blok atau per plot utama:

```
deggust::autoplot(desSPRAKL)+facet_wrap(~blok)
```

Oats

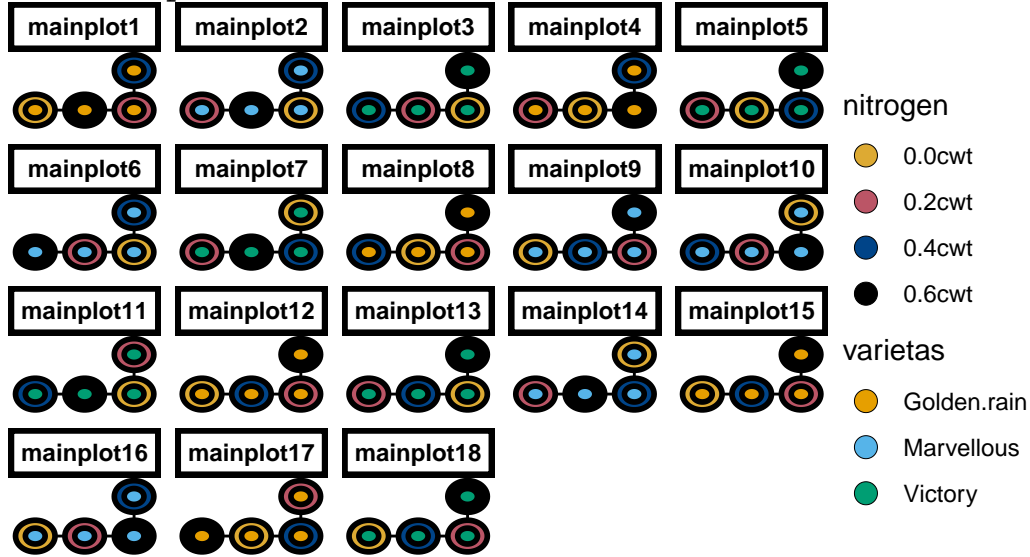
Unit: subplot



```
deggust::autoplot(desSPRAKL)+facet_wrap(~mainplot)
```


Oats

Unit: subplot



6.2.2 ANOVA

Model linear aditif bagi split plot RAL adalah:

$$y_{ijk} = \mu + \alpha_i + K_k + \delta_{k(i)} + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Dengan:

1. y_{ijk} adalah nilai pengamatan pada faktor A taraf ke-i dan faktor B taraf ke-j pada blok ke-k.
2. μ komponen aditif dari rata-rata umum.
3. α_i pengaruh utama faktor A. Diasumsikan $\sum_{i=1}^a \alpha_i = 0$ bagi model tetap dan $\alpha \sim N(0, \sigma_a^2)$ bagi model acak.
4. β_j pengaruh utama faktor B. Diasumsikan $\sum_{j=1}^b \beta_j = 0$ bagi model tetap dan $\beta \sim N(0, \sigma_b^2)$ bagi model acak.
5. K_k pengaruh utama kelompok.
6. $(\alpha\beta)_{ij}$ komponen interaksi faktor A dan faktor B. Diasumsikan $\sum_{i=1}^p \sum_{j=1}^t (\alpha\beta)_{ij} = 0$ bagi model tetap dan $(\alpha\beta)_{ij} \sim N(0, \sigma_{ab}^2)$ bagi model acak.
7. $\delta_{k(i)} \sim N(0, \sigma_\delta^2)$ komponen acak dari petak utama.
8. $\epsilon_{ijk} \sim N(0, \sigma^2)$ komponen acak anak petak.

Dan hipotesis pengaruh petak utama:

$$H_0 : \alpha_1 = \dots = \alpha_a = 0$$

faktor A tidak berpengaruh pada respon

$$H_1 : \text{Paling sedikit ada satu } i \text{ di mana } \alpha_i \neq 0$$

Hipotesis anak petak:

$$H_0 : \beta_1 = \dots = \beta_b = 0$$

faktor B tidak berpengaruh pada respon

$$H_1 : \text{Paling sedikit ada satu } j \text{ di mana } \beta_j \neq 0$$

Hipotesis interaksi:

$$H_0 : (\alpha\beta)_{11} = (\alpha\beta)_{12} = \dots = (\alpha\beta)_{ab} = 0$$

interaksi tidak berpengaruh pada respon

$$H_1 : \text{Paling sedikit ada sepasang } ij \text{ di mana } (\alpha\beta)_{ij} \neq 0$$

Serta pengaruh kelompok:

$$H_0 : K_1 = \dots = K_k = 0$$

Kelompok tidak berpengaruh pada respon

$$H_1 : \text{Paling sedikit ada satu } k \text{ di mana } K_k \neq 0$$

Dan tabel ANOVA:

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
Blok	r-1	JKK	JKK/dbA	CTK/CTGA	
A	a-1	JKA	JKA/dbA	CTA/CTGA	
Galat (a)	(a-1)(r-1)	JKGA	JKGA/dbGA		
B	(b-1)	JKB	JKB/dbB	CTB/CTGB	
AB	(a-1)(b-1)	JKAB	JKAB/dbAB	CTAB/CTGB	
Galat (b)	(a-1)(b-1)(r-1)	JKGB	JKGB/dbGB		
Total	abr-1	JKT			

Jumlah kuadrat tersebut dihitung dengan:

$$\begin{aligned}
FK &= \frac{y_{...}^2}{abr} \\
JKT &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (y_{ijk} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r y_{ijk}^2 - FK \\
JKST &= \sum_{i=1}^a \sum_{k=1}^r (\bar{y}_{i.k} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{k=1}^r \frac{y_{i.k}^2}{b} - FK \\
JKA &= \sum_{i=1}^a (\bar{y}_{i..} - \bar{y}_{...})^2 = \sum_{i=1}^a \frac{y_{i..}^2}{br} - FK \\
JKK &= \sum_{k=1}^r (\bar{y}_{..k} - \bar{y}_{...})^2 = \sum_{k=1}^r \frac{y_{..k}^2}{ab} - FK \\
JKG_a &= JKST - JKA - JKK \\
JKB &= \sum_{j=1}^b (\bar{y}_{.j.} - \bar{y}_{...})^2 = \sum_{j=1}^b \frac{y_{.j.}^2}{ar} - FK \\
JKP &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \frac{y_{ij.}^2}{r} - FK \\
JKAB &= \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b (\bar{y}_{ij.} - \bar{y}_{...})^2 - JKA - JKB \\
&= JKP - JKB - JKA \\
JKG_b &= JKT - JKP - JKG_a
\end{aligned}$$

6.2.3 Analisis menggunakan R

Sama seperti Split Plot RAL, perlu ditambah sintaks `Error(Blok:Utama)` di `aov`:

```
aovSPloTRAK1<-aov(Y~V*N+B+Error(B:V),data=oats)
```

Warning in `aov(Y ~ V * N + B + Error(B:V), data = oats)`: `Error()` model is singular

```
summary(aovSPloTRAK1)
```

Error: B:V

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
--	----	--------	---------	---------	--------

```

V          2    1786      893    1.485 0.2724
B          5   15875     3175    5.280 0.0124 *
Residuals 10    6013      601
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
      Df Sum Sq Mean Sq F value    Pr(>F)
N         3  20020     6673  37.686 2.46e-12 ***
V:N        6    322       54   0.303   0.932
Residuals 45   7969      177
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Dapat juga digunakan lme4:

```

library(lme4)
fit.oats <- lmer(Y ~ B + V * N + (1 | B:V), data = oats)
knitr::kable(anova(fit.oats))

```

	npar	Sum Sq	Mean Sq	F value
B	5	4675.0427	935.0085	5.2800486
V	2	526.0578	263.0289	1.4853399
N	3	20020.5000	6673.5000	37.6856497
V:N	6	321.7500	53.6250	0.3028236

Nilai F dan derajat bebas dari kedua tabel ANOVA sama, tetapi jumlah kuadrat berbeda. Fungsi `sp.plot` dari library `agricolae` juga dapat digunakan untuk menghitung tabel ANOVA tersebut. Sintaks metode tersebut adalah `with(data, sp.plot(blok, petak utama, anak petak, Y))`.

```

with(oats,sp.plot(block=B, pplot=V, splot=N, Y=Y))

```

ANALYSIS SPLIT PLOT: Y
Class level information

```

V   : Victory Golden.rain Marvellous
N   : 0.0cwt 0.2cwt 0.4cwt 0.6cwt
B   : I II III IV V VI

```

Number of observations: 72

Analysis of Variance Table

Response: Y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
B	5	15875.3	3175.1	NaN	NaN
V	2	1786.4	893.2	1.4853	0.2724
Ea	10	6013.3	601.3	NaN	NaN
N	3	20020.5	6673.5	37.6856	2.458e-12 ***
V:N	6	321.8	53.6	0.3028	0.9322
Eb	45	7968.7	177.1	NaN	NaN

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

cv(a) = 23.6 %, cv(b) = 12.8 %, Mean = 103.9722

Jumlah kuadrat dan kuadrat tengah galat sama, tetapi metode tersebut tidak dapat menguji pengaruh blok.

Karena setidaknya salah satu pengaruh konsentrasi pupuk nitrogen beda dari nol dan taraf faktor tersebut kuantitatif dengan jarak sama, dapat digunakan polinomial orthogonal sebagai uji lanjut. Polinomial orthogonal dapat dimasukkan dengan manual atau dengan `contr.poly`:

```
#contr.poly:
contrasts(oats$N)<-contr.poly(levels(oats$N))
aovSPloTRAK1<-aov(Y~V*N+B+Error(B:V),data=oats)
```

Warning in aov(Y ~ V * N + B + Error(B:V), data = oats): Error() model is singular

```
summary(aovSPloTRAK1,split=list(N=list("Linear"=1, "Kuadratik"=2,
                                         "Kubik"=3)))
```

Error: B:V

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
V	2	1786	893	1.485	0.2724
B	5	15875	3175	5.280	0.0124 *
Residuals	10	6013	601		

```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
      Df Sum Sq Mean Sq F value    Pr(>F)
N      3  20020     6674   37.686 2.46e-12 ***
  N: Linear      1  19536     19536 110.323 1.09e-13 ***
  N: Kuadratik   1    480        480   2.713   0.106
  N: Kubik       1     4         4    0.020   0.887
V:N      6    322        54   0.303   0.932
  V:N: Linear    2    168        84   0.475   0.625
  V:N: Kuadratik 2     11         6   0.031   0.969
  V:N: Kubik     2    142        71   0.402   0.671
Residuals    45   7969        177
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Terlihat bahwa pengaruh konsentrasi pupuk nitrogen linear setelah diuji melalui polinomial orthogonal.

6.3 Latihan Soal

1. Petani John. Sumber: ANOVA and Mixed Models: A Short Intro Using R (Lukas Meier, 2018)

Petani John memiliki delapan petal tanah yang berbeda. Dia mengacak dan menerapkan dua “skema” pemupukan (“kontrol” dan “baru”) dengan cara yang seimbang ke delapan plot. Selain itu, setiap plot dibagi menjadi empat subplot. Di setiap plot, empat varietas stroberi yang berbeda diacak ke subplot. John tertarik pada efek skema pemupukan dan varietas stroberi pada massa buah. Per subplot, ia mencatat massa buah setelah jangka waktu tertentu. Buat bagan percobaan dan analisis datanya!

Data dapat diakses dengan sintaks R berikut:

```

book.url <- "http://stat.ethz.ch/~meier/teaching/book-anova"
john <- readRDS(url(file.path(book.url, "data/john.rds")))
summary(john)

```

	plot	fertilizer	variety	mass
1	:4	control:16	A:8	Min. : 7.70
2	:4	new :16	B:8	1st Qu.:11.57

3	:4	C:8	Median :14.30
4	:4	D:8	Mean :13.94
5	:4		3rd Qu.:16.75
6	:4		Max. :20.60
(Other)	:8		

2. Benih

Dalam suatu percobaan tanaman, seorang petani menggunakan percobaan 2 faktor untuk melihat bagaimana kedua faktor tersebut mempunyai pengaruh terhadap pertumbuhan serta hasil tanaman, Untuk faktor pertama petani menggunakan (Takaran pupuk kandang: K1, K2, K3) diletakkan sebagai petak utama, sedangkan untuk faktor kedua (Jumlah benih per lubang: J1, J2, J3) dianggap sebagai anak petak. Setiap perlakuan diulang sebanyak 3 kali, dan unit-unit percobaan diasumsikan homogen.

Buat bagan dan analisis datanya!

3. Buat plot interaksi untuk data oats! Pisahkan plot tersebut menjadi *facet* untuk tiap blok.

7 Rancangan Split Blok

Filosofi dasar rancangan split blok mirip dengan strip plot - ada petak utama dan anak petak. Anak petak tersebut mengandung suatu aspek dari percobaan yang ingin ditekankan (diketahui pengaruhnya dengan ketepatan lebih tinggi). Dalam split blok kasus ini yang ditekankan bukan satu faktor, melainkan interaksi antara faktor. Oleh karena itu, faktor menjadi petak utama.

Rancangan split blok hanya dapat diterapkan dalam RAKL, tidak di RAL. Unit percobaan harus dipisahkan menjadi blok, lalu petak utama diacak di lajur dan baris blok tersebut. Pengacakan semua faktor di lajur dan baris tidak mungkin terjadi di satu unit percobaan yang hanya mengandung satu perlakuan atau satu kombinasi faktor sehingga dibutuhkan blok agar pengacakan semua kombinasi tersebut dapat dilakukan.

7.1 Pengacakan

Langkah pengacakannya adalah sebagai berikut:

- Pilihlah kelompok unit percobaan secara acak
- Tempatkan taraf-taraf faktor A secara acak pada setiap kelompok mengikuti plot lajur
- Tempatkan taraf-taraf faktor B secara acak pada setiap kelompok mengikuti plot baris.

7.1.1 Pengacakan di agricolae

Sebelum langkah-langkah pengacakan di atas diikuti, akan didefinisikan faktor yang akan dicobakan dan tarafnya. Hendak dicobakan pupuk dengan taraf “Kontrol”, “PK”, “N”, “NP”, “NK”, dan “NPK” serta genotipe beras dengan taraf IR-64 dan S-969.

```
P<-c("Kontrol", "PK", "N", "NP", "NK", "NPK")
G<-c("IR-64", "S-969")
```

Pengacakan split blok di agricolae cukup sederhana. Nama lain dari split blok adalah strip-plot, dan ada fungsi `design.strip(faktor1, faktor2, nblok)` di package tersebut:

```
baganSplitB<-agricolae::design.strip(P,G,r=4,serie=1)
```


Perlihatkan bagan percobaan:

```
knitr::kable(head(baganSplitB$book, n=10),
               col.names= c("Petak", "Blok",
                             "Pupuk", "Genotipe"))
```

Petak	Blok	Pupuk	Genotipe
11	1	NPK	S-969
12	1	NPK	IR-64
13	1	NP	S-969
14	1	NP	IR-64
15	1	PK	S-969
16	1	PK	IR-64
17	1	Kontrol	S-969
18	1	Kontrol	IR-64
19	1	N	S-969
20	1	N	IR-64

7.1.2 Pengacakan di edibble

Dalam rancangan strip plot, ada empat unit percobaan. Pengaruh blok tentu diaplikasikan ke blok, lalu pengaruh faktor ditaruh di lajur dan baris, serta interaksi di tiap unit. Maka, di edibble buat empat unit. Blok, lalu kolom dan baris yang `nested_in(blok)`, dan petak yang di dalam blok - `nested_in(blok)` dan `crossed_by(baris, kolom)` - tiap petak merupakan gabungan dari baris dan kolom. Cara mendefinisikan kasus ini adalah dengan menyatakan bahwa petak `crossed_by()` terlebih dahulu, lalu `nested_in()` - di tiap blok, petak merupakan gabungan baris dan kolom. Karena operasi dalam kurung dilaksanakan terlebih dahulu, bentuknya menjadi `nested_in(blok, crossed_by())`.

Lalu, pengacakan dilakukan ke baris dan kolom.

```
library(edibble)

desSBlok<-design(name="Padi") %>%
  set_units(blok=4,
            baris=nested_in(blok,2),
            kolom=nested_in(blok,6),
            petak=nested_in(blok,crossed_by(baris,kolom))) %>%
  set_trts(Genotipe=G,
           Pupuk=P) %>%
```

```
allot_trts(Genotype~baris,
           Pupuk~kolom) %>%
assign_trts("random", seed=420) %>% serve_table

knitr::kable(head(desSBlok,n=10))
```

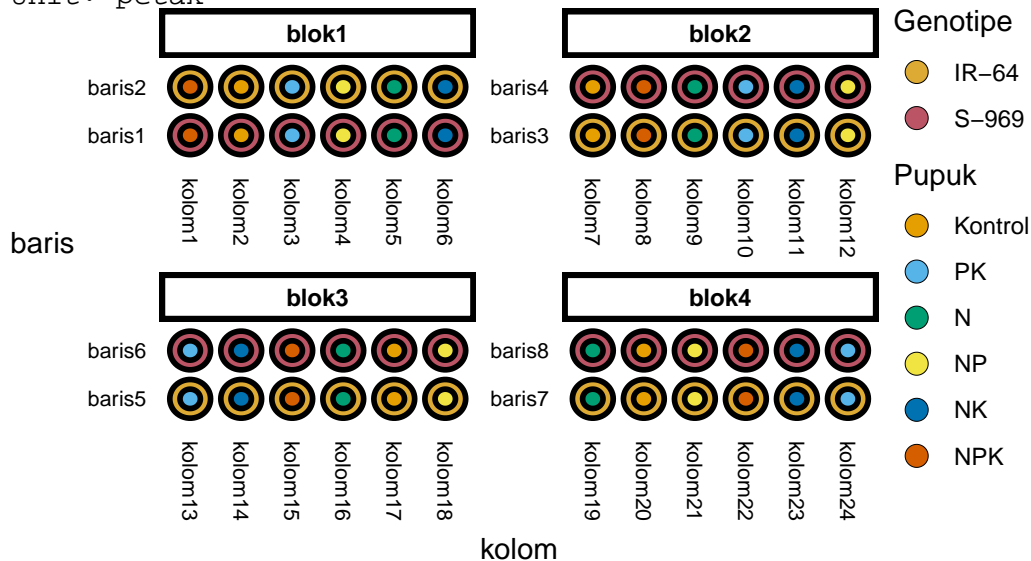
blok	baris	kolom	petak	Genotype	Pupuk
blok1	baris1	kolom1	petak1	S-969	NPK
blok1	baris2	kolom1	petak2	IR-64	NPK
blok1	baris1	kolom2	petak3	S-969	Kontrol
blok1	baris2	kolom2	petak4	IR-64	Kontrol
blok1	baris1	kolom3	petak5	S-969	PK
blok1	baris2	kolom3	petak6	IR-64	PK
blok1	baris1	kolom4	petak7	S-969	NP
blok1	baris2	kolom4	petak8	IR-64	NP
blok1	baris1	kolom5	petak9	S-969	N
blok1	baris2	kolom5	petak10	IR-64	N

Plot rancangan tersebut:

```
deggust::autoplot(desSBlok)
```

Padi

Unit: petak



Bandingkan dengan default edible:

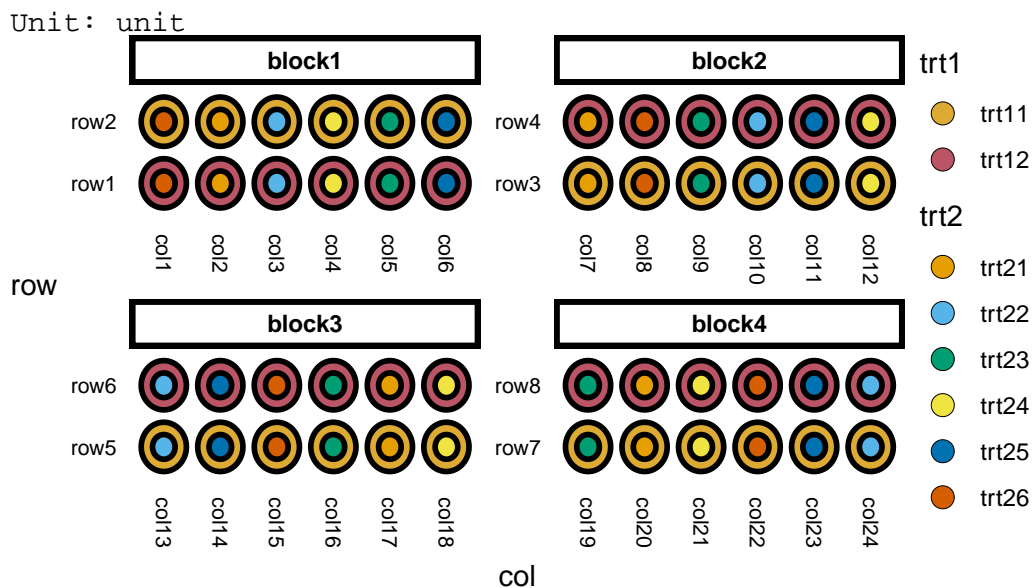
```
strip <- takeout(menu_strip(t1 = 2, t2 = 6, r = 4, seed=420))
examine_recipe(strip)
```

```
design("Strip-Plot Design | Strip-Unit Design") %>%
  set_units(block = 4,
            row = nested_in(block, 2),
            col = nested_in(block, 6),
            unit = nested_in(block, crossed_by(row, col))) %>%
  set_trts(trt1 = 2,
          trt2 = 6) %>%
  allot_trts(trt1 ~ row,
            trt2 ~ col) %>%
  assign_trts("random", seed = 420) %>%
  serve_table()
```

Plotnya akan relatif sama:

```
deggust::autoplot(strip)
```

Strip-Plot Design | Strip-Unit Design



7.2 ANOVA

7.2.1 Teori

Model linear aditif bagi split plot RAL adalah:

$$y_{ijk} = \mu + \alpha_i + K_k + \delta_{ik} + \beta_j + \gamma_{jk} + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Dengan:

1. y_{ijk} adalah nilai pengamatan pada faktor A taraf ke-i dan faktor B taraf ke-j pada blok ke-k.
2. μ komponen aditif dari rata-rata umum.
3. α_i pengaruh utama faktor A. Diasumsikan $\sum_{i=1}^a \alpha_i = 0$ bagi model tetap dan $\alpha \sim N(0, \sigma_a^2)$ bagi model acak.
4. β_j pengaruh utama faktor B. Diasumsikan $\sum_{j=1}^b \beta_j = 0$ bagi model tetap dan $\beta \sim N(0, \sigma_b^2)$ bagi model acak.
5. K_k pengaruh utama kelompok.
6. $(\alpha\beta)_{ij}$ komponen interaksi faktor A dan faktor B. Diasumsikan $\sum_{i=1}^p (\alpha\beta)_{ij} = \sum_{j=1}^t (\alpha\beta)_{ij} = 0$ bagi model tetap dan $(\alpha\beta)_{ij} \sim N(0, \sigma_{ab}^2)$ bagi model acak.
7. $\delta_{ik} \sim N(0, \sigma_\delta^2)$ komponen acak dari faktor A.
8. $\gamma_{jk} \sim N(0, \sigma_\gamma^2)$ komponen acak dari faktor B.
9. $\epsilon_{ijk} \sim N(0, \sigma^2)$ komponen acak interaksi dari faktor A dan B.

Dan hipotesis pengaruh faktor petak utama:

$$\begin{aligned} H_0 : \alpha_1 = \dots = \alpha_a = 0 \\ \text{faktor A tidak berpengaruh pada respon} \\ H_1 : \text{Paling sedikit ada satu } i \text{ di mana } \alpha_i \neq 0 \end{aligned}$$

Hipotesis pengaruh faktor anak petak:

$$\begin{aligned} H_0 : \beta_1 = \dots = \beta_b = 0 \\ \text{faktor B tidak berpengaruh pada respon} \\ H_1 : \text{Paling sedikit ada satu } j \text{ di mana } \beta_j \neq 0 \end{aligned}$$

Hipotesis pengaruh interaksi:

$$H_0 : (\alpha\beta)_{11} = (\alpha\beta)_{12} = \dots = (\alpha\beta)_{ab} = 0$$

interaksi tidak berpengaruh pada respon

$$H_1 : \text{Paling sedikit ada sepasang } ij \text{ di mana } (\alpha\beta)_{ij} \neq 0$$

Dan hipotesis pengaruh kelompok:

$$H_0 : K_1 = \dots = K_k = 0$$

Kelompok tidak berpengaruh pada respon

$$H_1 : \text{Paling sedikit ada satu } k \text{ di mana } K_k \neq 0$$

Dan tabel ANOVA:

Sumber Keragaman	db	JK	KT	F-hit	F(dbP,dbG)
Blok	r-1	JKK	JKK/dbA	GTK/KTGA	
A	a-1	JKA	JKA/dbA	KA/KTGA	
Galat (a)	(a-1)(r-1)	JKA	JKA/dbA	KA/KTGA	
B	(b-1)	JKB	JKB/dbB	KB/KTGB	
Galat (b)	(b-1)(r-1)	JKB	JKB/dbB	KB/KTGB	
AB	(a-1)(b-1)	JKAB	JKAB/dbAB	KTAB/KTGC	
Galat (c)	(a-1)(b-1)(r-1)	JKGC	JKGC/dbGC		
Total	abr-1	JKT			

Untuk bagian pertama tabel, jumlah kuadrat dihitung dengan:

$$FK = \frac{y_{...}^2}{abr}$$

$$JKT = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (y_{ijk} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r y_{ijk}^2 - FK$$

$$JKST_a = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{i..} - \bar{y}_{...})^2 = \sum_{i=1}^a \sum_{k=1}^r \frac{y_{i..}^2}{b} - FK$$

$$JKA = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{i..} - \bar{y}_{...})^2 = \sum_{i=1}^a \frac{y_{i..}^2}{br} - FK$$

$$JKB = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{.jk} - \bar{y}_{...})^2 = \sum_{k=1}^r \frac{y_{.k}^2}{ab} - FK$$

$$JKG_a = JKST_a - JKA - JKB$$

Untuk bagian kedua tabel:

$$\begin{aligned}
 JKB &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{.j.} - \bar{y}_{...})^2 = \sum_{j=1}^b \frac{y_{.j.}^2}{ar} - FK \\
 JKST_b &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{ij.} - \bar{y}_{...})^2 = \sum_{k=1}^r \sum_{j=1}^b \frac{y_{.jk}^2}{a} - FK \\
 JKG_b &= JKST_b - JKB
 \end{aligned}$$

Dan untuk bagian terakhir tabel:

$$\begin{aligned}
 JKP &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{ij.} - \bar{y}_{..})^2 = \sum_{i=1}^a \sum_{j=1}^b \frac{y_{ij.}^2}{r} - FK \\
 JKAB &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2 \\
 &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^r (\bar{y}_{ij.} - \bar{y}_{...})^2 - JKA - JKB \\
 &= JKP - JKB - JKA \\
 JKG_c &= JKT - JKP - JKK - JKG_a - JKG_b
 \end{aligned}$$

7.2.2 Analisis dengan R

Load data:

```
library(google sheets4)
library(data.table)

Data.SB<-as.data.table(read_sheet("https://docs.google.com/spreadsheets/d/1meWerIyAv1X8jtH
```

! Using an auto-discovered, cached token.

To suppress this message, modify your code or options to clearly consent to the use of a cached token.

See gargle's "Non-interactive auth" vignette for more details:

<<https://gargle.r-lib.org/articles/non-interactive-auth.html>>

i The googlesheets4 package is using a cached token for 'muh.ammarsahab@apps.ipb.ac.id'.

v Reading from "Data P10 STA1222".

v Range 'Sheet1'.

```
knitr::kable(Data.SB)
```

P	G	B1	B2	B3	B4
Kontrol	IR-64	20.7	32.1	29.5	37.7
Kontrol	S-969	27.7	33.0	26.3	37.7
PK	IR-64	30.0	30.7	25.5	36.9
PK	S-969	36.6	33.8	27.0	39.0
N	IR-64	39.9	41.5	46.4	44.5
N	S-969	37.4	41.2	45.4	44.6
NP	IR-64	40.8	43.5	43.3	43.4
NP	S-969	42.2	46.0	45.9	46.2
NK	IR-64	42.4	45.6	44.8	47.0
NK	S-969	39.8	39.5	40.9	44.0
NPK	IR-64	48.6	49.8	42.6	46.6
NPK	S-969	42.9	45.9	43.9	45.6

Karena data belum berbentuk *long* - satu baris adalah satu observasi, melt data tersebut. Tiap baris dalam data awal diidentifikasi oleh kombinasi pupuk dan genotipe, sehingga `id.vars` adalag `c("P","G")`:

```
library(reshape2)

Data.SB<-melt(
  Data.SB,
  id.vars=c("P","G"),
  value.name = "H"
)

knitr::kable(head(Data.SB,10))
```

P	G	variable	H
Kontrol	IR-64	B1	20.7
Kontrol	S-969	B1	27.7
PK	IR-64	B1	30.0
PK	S-969	B1	36.6
N	IR-64	B1	39.9
N	S-969	B1	37.4
NP	IR-64	B1	40.8
NP	S-969	B1	42.2
NK	IR-64	B1	42.4
NK	S-969	B1	39.8

Ubah kolom `variable` menjadi blok:

```
library(data.table)
```

Attaching package: 'data.table'

The following objects are masked from 'package:reshape2':

`dcast`, `melt`

```
setnames(Data.SB,"variable", "B")
```

Untuk melakukan ANOVA, perlu ada struktur error yang khusus. Pada dasarnya, akan ada error khusus untuk faktor serta interaksinya, sehingga tulis `Error(B:(P*G))` - akan dihitung `Error(B:P)`, `Error(B:G)`, dan `Error(B:(G:P))`, atau error tiap faktor dan interaksinya di tiap ulangan (blok). Spesifikasi lain adalah:

```
aov.SB<-aov(H~P*G+B+Error(B:(P*G)),Data.SB)
```

Warning in `aov(H ~ P * G + B + Error(B:(P * G)), Data.SB)`: `Error()` model is singular

```
summary(aov.SB)
```


Error: B:P

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
P	5	1674.8	335.0	18.767	5.62e-06 ***
B	3	197.1	65.7	3.681	0.0362 *
Residuals	15	267.7	17.8		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: B:G

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
G	1	0.035	0.0352	0.032	0.87
Residuals	3	3.329	1.1097		

Error: B:P:G

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
P:G	5	78.59	15.72	4.504	0.0105 *
Residuals	15	52.35	3.49		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
aov.SB1<-aov(H~P*G+B+Error(B:P*B:G),Data.SB)
```

Warning in aov(H ~ P * G + B + Error(B:P * B:G), Data.SB): Error() model is singular

```
summary(aov.SB1)
```

Error: B:P

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
P	5	1674.8	335.0	18.767	5.62e-06 ***
B	3	197.1	65.7	3.681	0.0362 *
Residuals	15	267.7	17.8		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: B:G

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
G	1	0.035	0.0352	0.032	0.87

```
Residuals 3 3.329 1.1097
```

```
Error: B:P:G
```

```
      Df Sum Sq Mean Sq F value Pr(>F)
P:G      5  78.59   15.72   4.504 0.0105 *
Residuals 15  52.35    3.49
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pastikan ada kurung, karena anova akan salah tanpa kurung:

```
aov.SB1<-aov(H~P*G+B+Error(B:P*G),Data.SB)
```

Warning in aov(H ~ P * G + B + Error(B:P * G), Data.SB): Error() model is singular

```
summary(aov.SB1)
```

```
Error: G
```

```
      Df Sum Sq Mean Sq
G      1 0.03521 0.03521
```

```
Error: B:P
```

```
      Df Sum Sq Mean Sq F value    Pr(>F)
P      5 1674.8   335.0  18.767 5.62e-06 ***
B      3  197.1    65.7   3.681  0.0362 *
Residuals 15  267.7    17.8
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Error: B:P:G
```

```
      Df Sum Sq Mean Sq F value    Pr(>F)
P:G      5  78.59   15.718   5.081 0.00444 **
Residuals 18  55.68    3.093
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Atau, dapat digunakan fungsi `strip.plot(blok, faktor 1, faktor 2, respon)` dari `agri-colae`:

```
library(agricolae)
```

```
out1<-with(Data.SB,strip.plot(B,P,G,H))
```

ANALYSIS STRIP PLOT: H

Class level information

P : Kontrol PK N NP NK NPK

G : IR-64 S-969

B : B1 B2 B3 B4

Number of observations: 48

model Y: H ~ B + P + Ea + G + Eb + G:P + Ec

Analysis of Variance Table

Response: H

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
B	3	197.11	65.70		
P	5	1674.80	334.96	18.7668	5.616e-06 ***
Ea	15	267.73	17.85		
G	1	0.04	0.04	0.0317	0.86997
Eb	3	3.33	1.11		
G:P	5	78.59	15.72	4.5038	0.01047 *
Ec	15	52.35	3.49		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

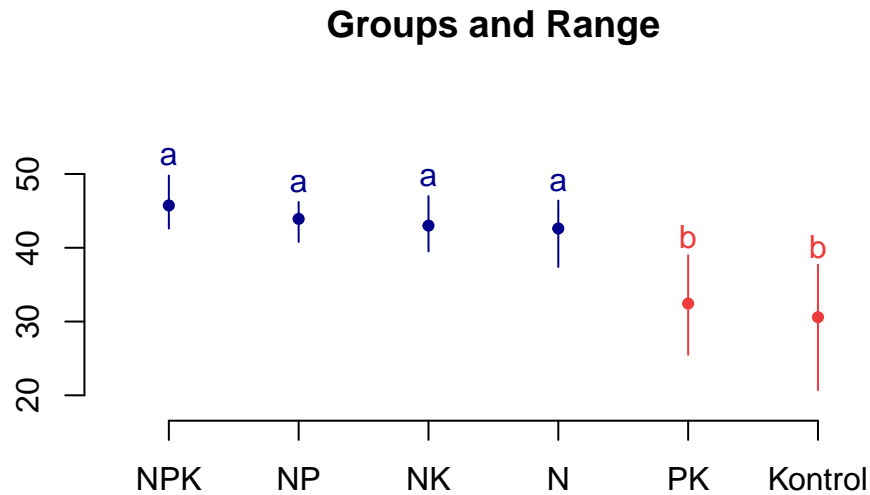
cv(a) = 10.6 %, cv(b) = 2.7 %, cv(c) = 4.7 %, Mean = 39.71458

Uji lanjut untuk tiap faktor dapat mudah dilakukan. gl.a adalah derajat bebas galat a, dan Ea galat A.

```
test1<-agricolae::LSD.test(Data.SB$H, Data.SB$P, out1$gl.a, out1$Ea)
test1$statistics
```

MSerror	Df	Mean	CV	t.value	LSD
17.84854	15	39.71458	10.63779	2.13145	4.502424

```
plot(test1)
```



Namun, tampaknya tidak dilakukan penyesuaian pada KTG sehingga perlu dilakukan perhitungan manual jika ingin ada penyesuaian tersebut:

7.2.3 Latihan: Huasahuasi

Sedang diteliti ambang batas curah hujan sebagai dukungan untuk aplikasi fungisida waktu dalam pengendalian penyakit busuk daun kentang di Peru menggunakan strip plot.

```
library(agricolae)

data(huasahuasi)
str(huasahuasi)
```

List of 2

```
$ AUDPC:'data.frame': 450 obs. of 13 variables:
..$ block: Factor w/ 3 levels "I","II","III": 1 1 1 1 1 1 1 1 1 1 ...
..$ trt : Factor w/ 3 levels "40mm","7-days",..: 1 1 1 1 1 1 1 1 1 1 ...
..$ clon : Factor w/ 5 levels "C386209.10","C387164.4",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```

..$ plant: num [1:450] 1 2 3 4 5 6 7 8 9 10 ...
..$ d44 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d51 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d58 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d65 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d72 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d79 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d86 : num [1:450] 0 0 0 0 0 0 0 0 0 0 ...
..$ d93 : num [1:450] 0 0 0 1 1 0 0 0 0 0 ...
..$ d100 : num [1:450] 0 0 0 1 1 0 0 0 0 0 ...
$ YIELD:'data.frame': 45 obs. of 6 variables:
..$ block: Factor w/ 3 levels "I","II","III": 1 1 1 1 1 1 1 1 1 1 ...
..$ trt : Factor w/ 3 levels "40mm","7-days",...: 1 1 1 1 1 2 2 2 2 2 ...
..$ clon : Factor w/ 5 levels "C386209.10","C387164.4",...: 2 1 3 5 4 2 1 3 5 4 ...
..$ y1da : num [1:45] 22.8 10.8 1.6 10.9 3.5 ...
..$ y2da : num [1:45] 5.5 8 11.7 9.9 5.1 7 12.5 9.15 8.8 4.6 ...
..$ y3da : num [1:45] 4.14 6.1 4.15 5.2 2.4 7.6 3.6 4.9 5.9 3.5 ...

```

Karena ada dua data.frame, ambil data.frame YIELD:

```

YIELD<-huasahuasi$YIELD
str(YIELD)

```

```

'data.frame': 45 obs. of 6 variables:
 $ block: Factor w/ 3 levels "I","II","III": 1 1 1 1 1 1 1 1 1 1 ...
 $ trt : Factor w/ 3 levels "40mm","7-days",...: 1 1 1 1 1 2 2 2 2 2 ...
 $ clon : Factor w/ 5 levels "C386209.10","C387164.4",...: 2 1 3 5 4 2 1 3 5 4 ...
 $ y1da : num 22.8 10.8 1.6 10.9 3.5 ...
 $ y2da : num 5.5 8 11.7 9.9 5.1 7 12.5 9.15 8.8 4.6 ...
 $ y3da : num 4.14 6.1 4.15 5.2 2.4 7.6 3.6 4.9 5.9 3.5 ...

```

Ada y1da, y2da, dan y3da. Respon yield merupakan penjumlahan dari ketiga respon tersebut
- `YIELD$yield<- y1da+y2da+y3da`, dan perlakuan ada di trt dan clon.

Buat bagan percobaan dengan edibble dan lakukan ANOVA!

8 ANCOVA

Terkadang, ada sumber variasi dalam percobaan yang dapat diamati secara kuantitatif, tetapi tidak dapat dikendalikan. Sumber variasi tersebut merupakan *peubah konkomitan*. Pada saat pengaruh sumber variasi tersebut diperhitungkan dalam analisis, kemungkinan:

1. Variasi peubah respon yang tidak dapat dijelaskan model berkurang, sehingga galat dapat dikendalikan.
2. Efek dari perlakuan dapat disesuaikan dengan peubah konkomitan tersebut.

8.1 Data

Untuk melihat pelaksanaan ANCOVA, akan digunakan studi kasus tersebut:

Suatu percobaan di bidang industri ingin mempelajari pengaruh mesin terhadap respons kekuatan serat yang dihasilkan yang dipergunakan dalam industri tekstil. Terdapat tiga perlakuan mesin yang dirancang dengan RAL, masing-masing diulang 5 kali. Telah diketahui bahwa kekuatan serat yang dihasilkan juga tergantung pada diameter serat tersebut. Untuk itu, dalam percobaan ini digunakan peubah penyerta (X) yaitu diameter serat yang dihasilkan. Respon kekuatan serat yang dihasilkan (Y) diukur dalam satuan tertentu. Diameter serat (X) diukur dalam 10^{-3} cm .

Semua analisis dimulai dengan membaca data. Sintaks pipa dari dplyr akan digunakan untuk memperjelas kode.

```
library(dplyr)

url<- "https://docs.google.com/spreadsheets/d/1YdRTf455ajZAc7--MdwYTqbCpGKmMkdmxIZcUXcwS3I

googlesheets4::read_sheet(url) %>%
  knitr::kable()
```

Ulangan	Y_Mesin1	X_Mesin1	Y_Mesin2	X_Mesin2	Y_Mesin3	X_Mesin3
1	36	20	40	22	35	21

Ulangan	Y_Mesin1	X_Mesin1	Y_Mesin2	X_Mesin2	Y_Mesin3	X_Mesin3
2	41	25	48	28	37	23
3	39	24	39	22	42	26
4	42	25	45	30	34	21
5	49	32	44	28	32	15

Dari output tabel, dapat dilihat suatu masalah di tabulasi data. Kita ingin tabel akhir memiliki 4 kolom, yaitu ulangan, mesin, X, dan Y. Beda dengan *reshaping* biasa yang menggabungkan beberapa kolom menjadi satu, kita harus menggabungkan kolom itu menjadi sejumlah kolom. **tidyr** memiliki metode *reshaping* yang lebih komprehensif dari **reshape2**, jadi akan dipakai library tersebut.

Fungsi **pivot_longer** mengubah data dari wide ke long. Dalam kasus ini, tiap kolom di data awal memiliki format X/Y_Mesin. Terlihat bahwa kolom tersebut dapat dipisah menjadi “X” atau “Y” dan “Mesin” menggunakan underscore. Tentu, yang kita inginkan adalah komputer menyadari kolom yang diawali “X” lalu dimasukkan ke kolom X di tabel akhir, begitu juga untuk Y. Argumen **names_sep** akan memisahkan X atau Y dengan keterangan mesin di underscore (_).

Lalu, satu kolom yang telah dipisahkan tersebut harus dimasukkan ke kolom baru. **names_to** menunjukkan kolom baru yang akan dimunculkan. Argumen **.value** memberi keterangan bahwa bagian pertama dari nama kolom data asli memberi keterangan kolom (X atau Y).

```
dat2 <- googlesheets4::read_sheet(url) %>%
  tidyr::pivot_longer(
    !Ulangan,
    names_to = c(".value", "Mesin"),
    names_sep = "_" ) %>%
  mutate(Mesin=stringr::str_sub(Mesin, -1))
```

v Reading from "ANCOVA - P11".

v Range 'Sheet1'.

Data sudah benar.

8.2 Visualisasi ANCOVA

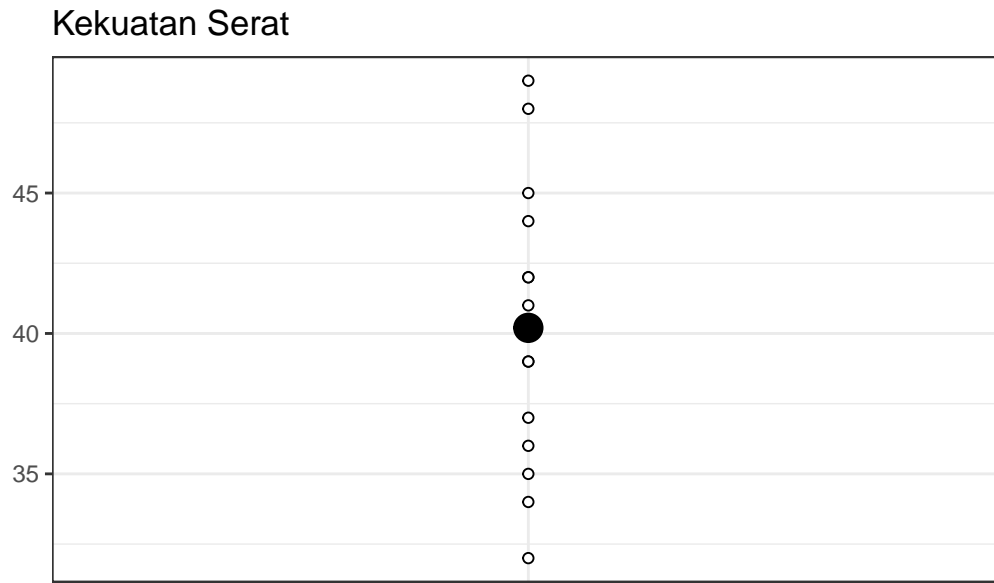
Sebelum melakukan perhitungan, kita dapat coba untuk memvisualisasi proses ANOVA dan ANCOVA. Pertama, mulai dengan data peubah respon saja, tanpa perlakuan ataupun peubah

konkomitan. Buat scatterplot (`geom_point`). Penduga untuk tiap observasi yang logis adalah rerata, yang dimunculkan dari `stat_summary(fun="mean")`.

```
library(ggplot2)

ggplot(data=dat2, aes(x="", y=Y))+
  geom_point(fill="#FFFFFF", shape=21)+
  stat_summary(fun=mean, size=1)+
  theme_bw()+
  xlab("")+ylab("")+
  ggtitle("Kekuatan Serat")
```

Warning: Removed 1 rows containing missing values (`geom_segment`).



Beberapa argumen di akhir kosmetik saja. `theme_bw()` membuat plot hitam-putih yang lebih rapih. Sekarang, tambahkan perlakuan. Tentu sekarang penduga logis bagi observasi di tiap perlakuan adalah rerata perlakuannya.

```
library(tayloRswift)

ggplot(data=dat2, aes(x=NA, y=Y, group=Mesin, colour=Mesin))+
```



```
geom_point(fill="#FFFFFF", shape=21)+
stat_summary(fun=mean, size=1)+
theme_bw()+
scale_color_taylor(palette="taylor1989")+
xlab("")+ylab("")+
ggtitle("Produksi per Mesin")
```

Warning: Removed 3 rows containing missing values (geom_segment).

