# Measuring IR Effectiveness

Alfan F. Wicaksono
Fakultas Ilmu Komputer, Universitas Indonesia

# Which one is better?

**Query**: cara mahasiswa bahagia

## SERP A

Google

cara mahasiswa bahagia

All | Videos | News | Images | Books | More | Tools

About 8,540,000 results (0.34 seconds)

**Rank 1**

https://www.beautynesia.id › life · Translate this page

Kiat Menjalani Kehidupan Kampus dengan Bahagia untuk ...

Sep 5, 2022 — Kiat Menjalani Kehidupan Kampus dengan **Bahagia** untuk Para **Mahasiswa**, Biar Lebih Bersemangat! · Bangun Hubungan Pertemanan yang Sefrekuensi · Buat ...

**Rank 2**

https://edukasi.okezone.com › read · Translate this page

Cara Jadi Mahasiswa Paling Bahagia - Okezone Edukasi

Mar 10, 2015 — **Cara** Jadi **Mahasiswa** Paling **Bahagia** · 1. Pintar · 2. Clubbing · 3. Bekerja · 4. Tangani stres.

**Rank 3**

https://www.kompasiana.com › Pendidikan › Pendidikan

Inilah Empat Jalan Bagi Mahasiswa agar Sukses Sekaligus ...

Aug 9, 2022 — Pahamilah bahwa kadangkala kita sendiri yang membuat rasa tidak **bahagia** itu. Penting bagi kita mempelajari, memahami, dan mengelola serta ...

**Rank 4**

https://itats.ac.id › kuliah-itu-bikin-ba... · Translate this page

KULIAH ITU BIKIN BAHAGIA - ITATS Institut Teknologi Adhi ...

Dec 23, 2018 — Bebas berdiskusi, saling berbagi pengetahuan dilakukan **mahasiswa** di segala penjuru kampus. Hobi pun tersalurkan dengan baik, **mahasiswa** dapat ...

**Rank 5**

https://www.idntimes.com › nunun-8 · Translate this page

Ngaku deh, 8 Hal Sederhana Ini Pasti Bikin Kamu yang ...

**SERP = Search Engine Results Page**

## SERP B

Microsoft Bing

cara mahasiswa bahagia

ALL | IMAGES | VIDEOS | MAPS | NEWS

200.000.000 Results | Date | Open links in new tab

**Rank 1**

Bahagia dalam Perspektif Mahasiswa - Kompasiana.com

www.kompasiana.com/pesonamu/5adcbc15ab12ae0762584e13/bahagia-dala...

Was this helpful?

**Rank 2**

Cara Jadi Mahasiswa Paling Bahagia : Okezone Edukasi

https://edukasi.okezone.com/read/2015/03/10/65/...

Web 10/03/2015 · JAKARTA - Bukan rahasia bila kehidupan perkuliahan penuh dengan stres.Selama empat tahun, kita akan disibukkan dengan berbagai tugas dan tanggung ...

**Rank 3**

Cara Asyik Jadi Mahasiswa Bahagia 1 : Okezone News

https://news.okezone.com/read/2014/06/27/373/...

Web Menjadi orang yang **bahagia** seharusnya adalah prioritas setiap **mahasiswa** Ini caranya - News Kampus - Okezone News

**Rank 4**

Kiat Menjalani Kehidupan Kampus dengan Bahagia untuk Para ...

https://www.beautynesia.id/life/kiat-menjalani...

Web 05/09/2022 · Kiat Menjalani Kehidupan Kampus dengan **Bahagia** untuk Para **Mahasiswa**, Biar Lebih Bersemangat! Kehidupan kampus bisa sangat menyenangkan ataupun ...

**Rank 5**

Cara Asyik Jadi Mahasiswa Bahagia 2 : Okezone News

https://news.okezone.com/read/2014/06/27/373/...

Web Orang bilang jadi **mahasiswa** juga penuh rasa stres Tetapi bisa kok tetap sibuk dengan seabrek kegiatan dan menjadi **mahasiswa** bahagianbsp - News Kampus - Okezone News

**Rank 6**

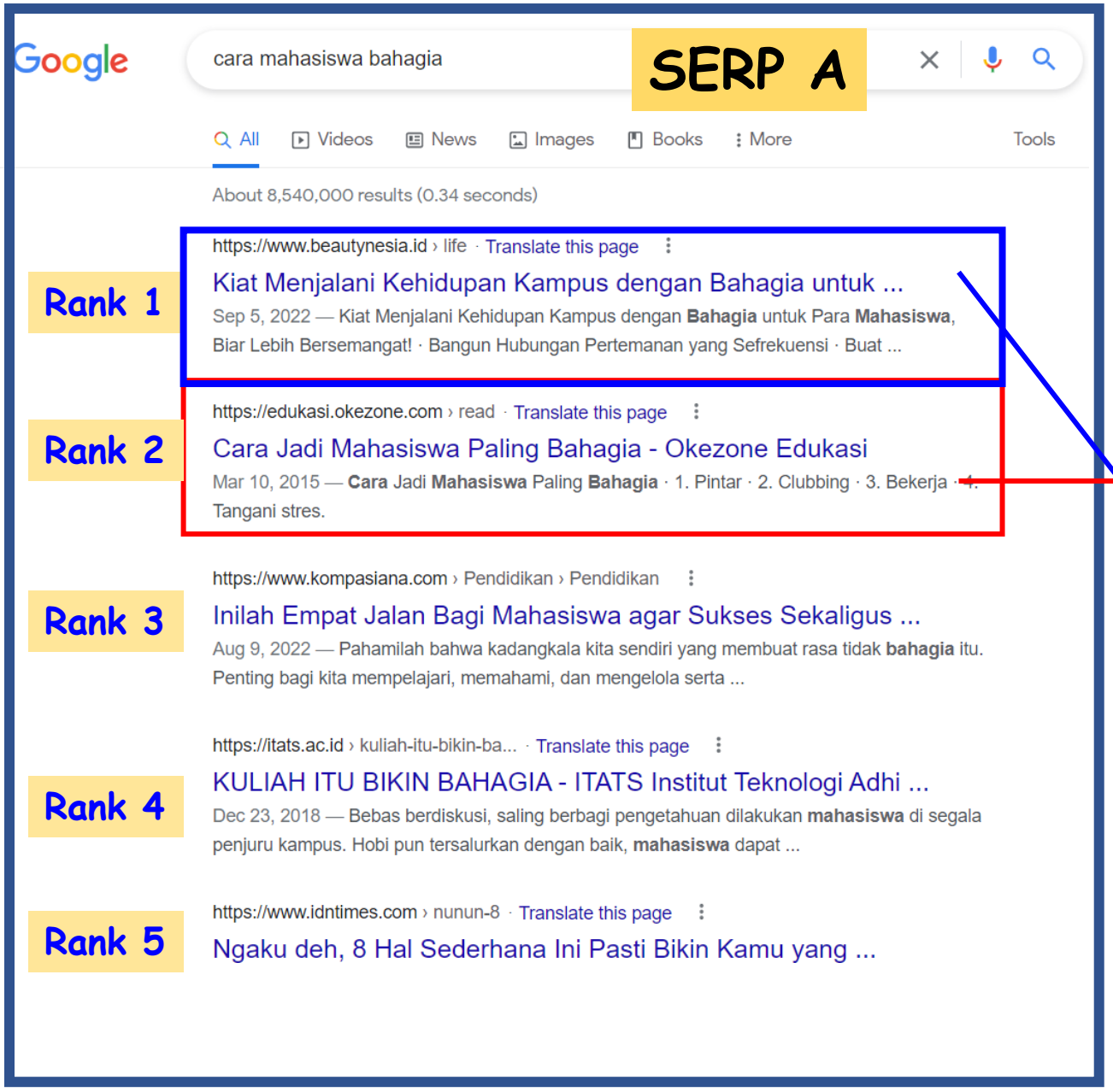
Tetap Bahagia Menjadi Mahasiswa Tingkat Akhir dengan ...

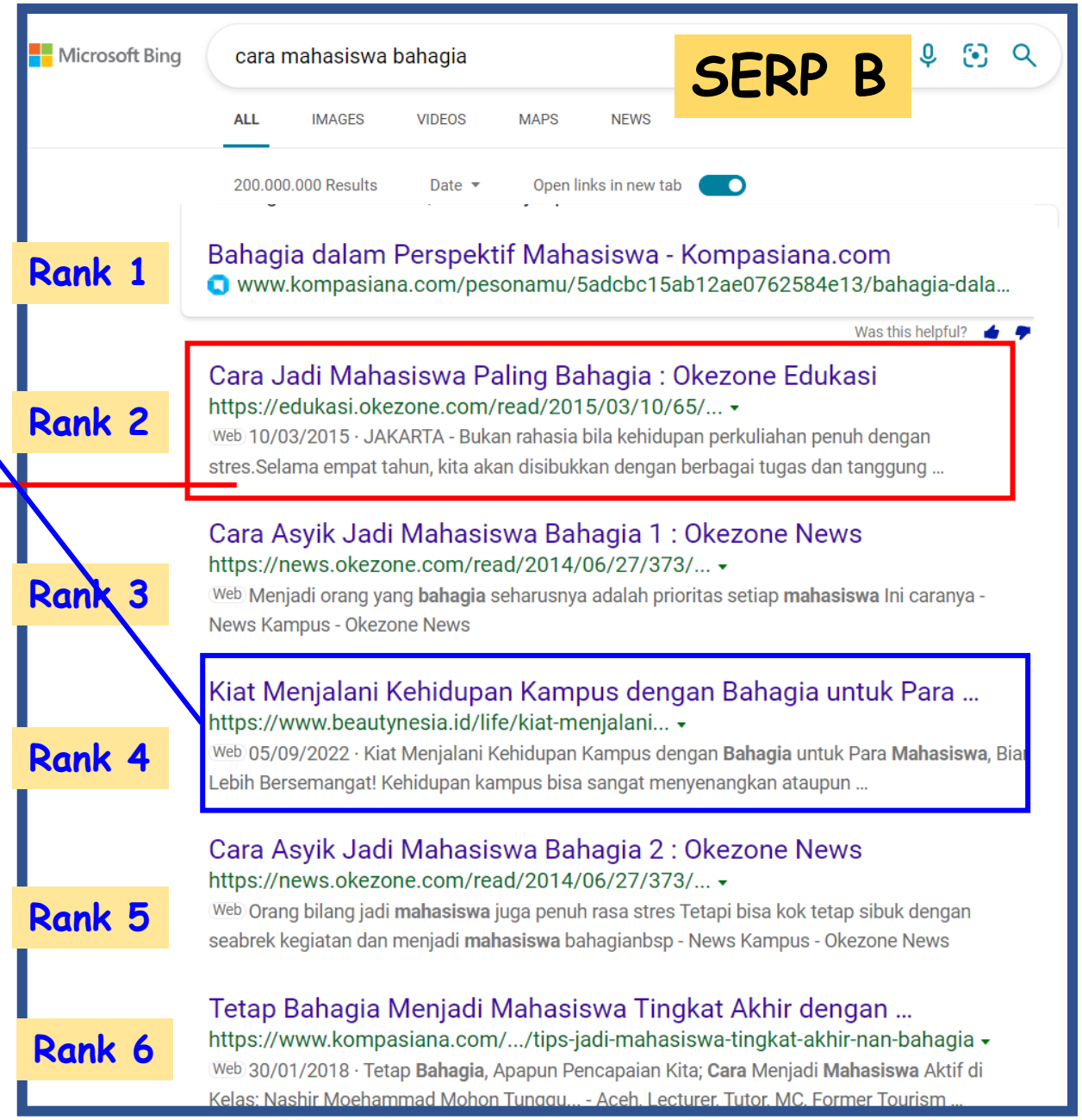
https://www.kompasiana.com/.../tips-jadi-mahasiswa-tingkat-akhir-nan-bahagia

Web 30/01/2018 · Tetap **Bahagia**, Apapun Pencapaian Kita; **Cara** Menjadi **Mahasiswa** Aktif di Kelas; Nashir Moehammad Mohon Tunggu... - Aceh. Lecturer. Tutor. MC. Former Tourism ...

# Which one is better?

**Query**: cara mahasiswa bahagia



# Search Engine Evaluation

- Online Evaluation (di "production environment" langsung)
  - A/B Testing
  - Interleaving

- Offline Evaluation (evaluasi di Lab)
  - User Study di Laboratorium
  - Evaluasi menggunakan **Test Collection**

# A/B Testing

A form of controlled experiment testing a causal relationship between system changes and their effects on the behaviour of users (clickthroughs and query reformulations).

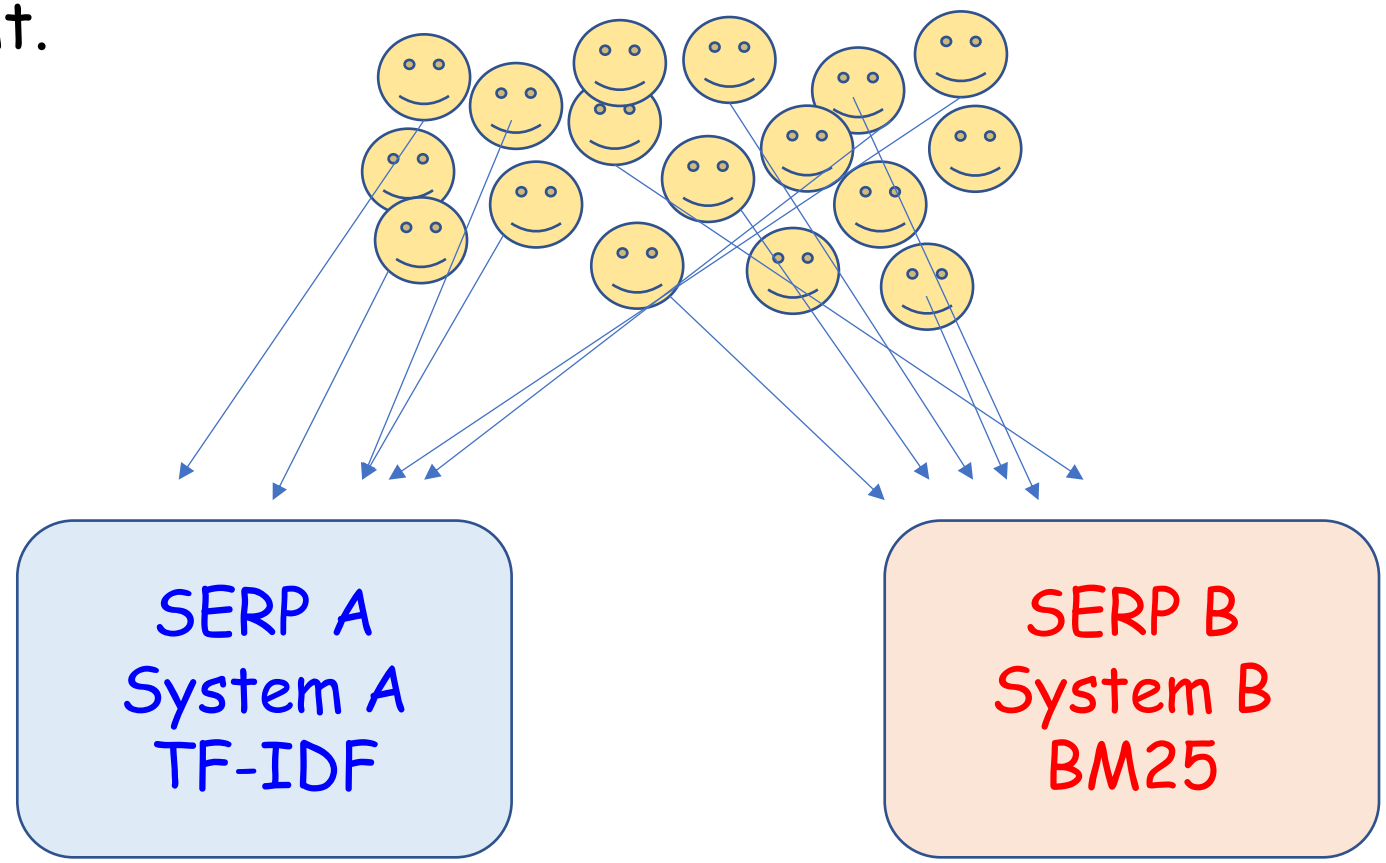
**Contoh**: saya ingin tahu mana yang lebih baik, ranking dengan TF-IDF atau dengan BM25?

SERP A
System A
TF-IDF

SERP B
System B
BM25

# A/B Testing

Assign them to a **large number of random users** in a production environment.

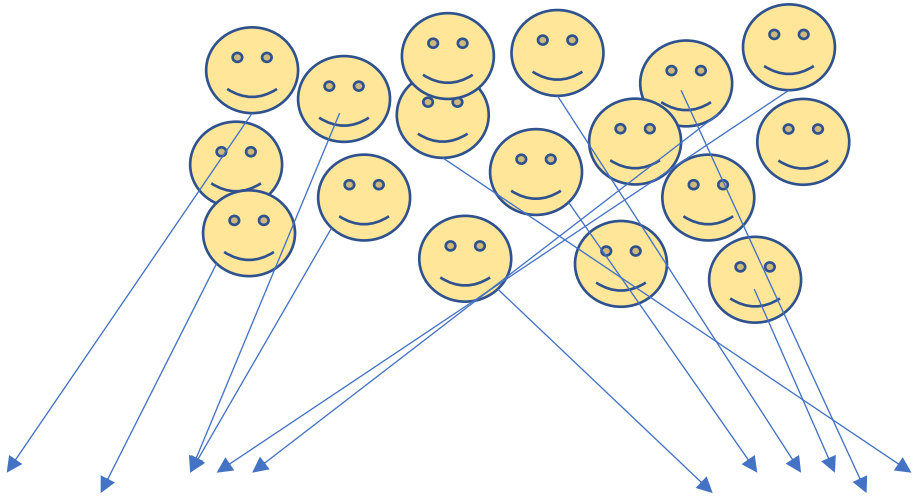


# A/B Testing

**Catatan:**
Clicks biasanya merupakan indikator **kepuasan** user.
Query reformulation merupakan indikator **ketidakpuasan**.

Assign them to a **large number of random users** in a production environment.

user gak tau (natural)



Observed Behavior per day:
**~5000 clicks**
**~1300 query reformulations**

SERP A
System A
TF-IDF

Observed Behavior per day:
**~100,000 clicks**
**~58 query reformulations**

SERP B
System B
BM25

karena kalau click biasanya user lebih puas, sedangkan query reformulation itu kayak user mau hasil yg lain.

**B is better**

# clicks received


SERP generated by System A

# clicks received

SERP generated by System B


**Implicit feedback seperti clickthrough rate biasanya digunakan untuk aproksimasi kepuasan user, dan menjadi basis untuk kebanyakan online evaluation**

# Interleaving

- Two rankings initiated from the same query are interleaved into a single ranked list using a certain strategy.
- The clickthrough information observed from the combined ranking is then used to decide which system provides better rankings.

Interleaving itu selang seling, jadi kayak hasil dari SERP A (TF-IDF) vs SERP B (IBM 25), hasilnya di interleave misal

| A | B |
| --- | --- |
| D68 (rank 1) | D53(rank1) |
| D69 (rank 2) | D90(rank2) |

Jadi hasilnya kalau di interleave jadinya:

D68
D53
D90
D69

ini bakal jadi hasilnya (ke interleave)

# Offline Evaluation

## Evaluation Based on Test Collection

Perangkat:

- A set of documents
  - $\{D_1, D_2, D_3, D_4, \ldots, D_M\}$
- A set of queries
  - $\{Q_1, Q_2, Q_3, Q_4, \ldots, Q_N\}$
- Relevance Judgments (**qrels**)
  - Biasanya biner, 1 jika relevan, dan 0 tidak

Jadi top M yang di evaluasi oleh manusia (offline evaluation). M itu hasil unionnya

Pooling

Misalnya dikasih suatu Query (namanya Q)

Ada 3 tim: A, B, dan C

Dan untuk model tim A top 3 nya:

D7 D8 D9

Untuk model tim B top3 nya:

D8 D10 D11

Untuk model tim C top 10nya:

D7, D8, 10

Nah semuanya ini itu di

gabung (UNION bukan intersect)

Hasil unionnya menjadi hal yang di judge oleh annotator

$Q_1\ D_1\ 0$
$Q_1\ D_2\ 1$
$Q_1\ D_3\ 1$
$Q_1\ D_4\ 0$
$Q_1\ D_5\ 1$
...
$Q_1\ D_M\ 0$
...
$Q_N\ D_1\ 1$
$Q_N\ D_2\ 1$
$Q_N\ D_3\ 0$
$Q_N\ D_4\ 0$
...

# Offline Evaluation

Perlu sebuah metric yang menilai kualitas dari sebuah ranking yang dihasilkan.

metric itu yang bakal dijlasin dibawah2 kayak DCG/Normalized DCG, Rank Biased Precision

```
scores = []
for each q in set_of_queries:
    results = retriever(q, set_of_docs)
    score = metric(results, q, qrels)
    scores.add(score)

overall_score = aggregate(scores)   // biasanya mean
```

Hasil yang diperoleh

Q -> D70 0
D80 1
D91 1

Nah Qreslnya itu (yang asli), query relevance

Q80 1
Q70 0
Q91 1
Q75 1
Q85 1

Dimana nanti dapet kalau Recall = 2/4 dan Precision = 2/3

4 itu dari berapa banyak query yang relevan di Qrels
3 itu banyaknya dokumen yang di retrieve yang match dengan boolean query

# Offline Evaluation Metrics: Boolean Retrieval

- **Precision**: dari himpunan dokumen yang di-retrieve, berapa proporsi yang benar-benar relevan?

$$Precision = \frac{\sum_{s \in S} rel(s)}{|S|}$$

- **Recall**: dari himpunan dokumen yang relevan (baik yang di-retrieve maupun yang tidak), ada berapa proporsi yang berhasil di-retrieve?

$$Recall = \frac{\sum_{s \in S} rel(s)}{|R|}$$

banyak yang gak pake F1 di IR lagi, karena gak jelas yang bikin jelek itu presisi atau recall

Jadi gak representatif, karena nilai tersebut dihasilkan karena kombinasi dari recall dan precision

**N** : himpunan dokumen di koleksi
**R** : himpunan dokumen di koleksi yang relevan
**S** : himpunan dokumen yang di-retrieve (match dengan Boolean query)

**rel(s)** : relevansi dari dokumen **s** (**1** relevan, **0** tidak)

# Metric untuk Ranked Retrieval?

Relevan

Tidak Relevan

**Query**: situs universitas jambi

Sebenarnya ranked retrieval itu dari behaviour usernya

## SERP A (Ranking A)

Yang bagus itu serp A karena behaviour orang hanya lihat 1/2 yang atas-atas aja. Karena yang pertama kali relevan ada di posisi 1

D14 D32 D41 D45 D47 D54 D70 D80 D71 D64

## SERP B (Ranking B)

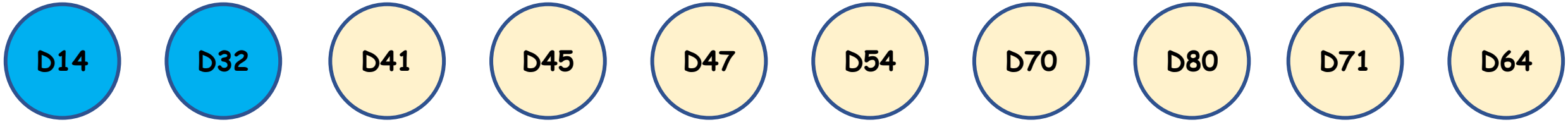
D10 D32 D14 D42 D46 D55 D69 D75 D91 D68

**Jika metric score antara 0 dan 1, berapa Anda berikan score untuk 2 ranking di atas?**

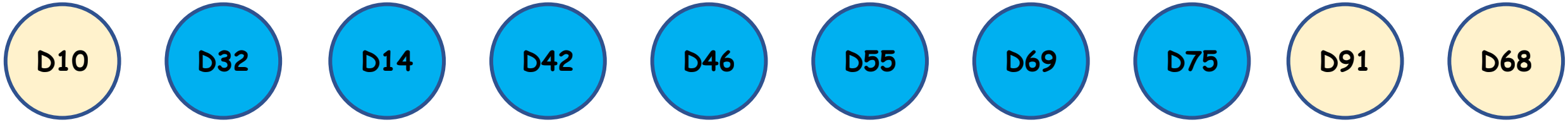
## Metric untuk Ranked Retrieval?


**Query**: situs universitas jambi

### SERP A (Ranking A)




### SERP B (Ranking B)




**Berapa banyak dokumen yang Anda lihat untuk memenuhi kebutuhan tersebut?**

# Metric untuk Ranked Retrieval?

Relevan

Tidak Relevan

**Query**: cara mahasiswa bahagia dan sukses

**Berapa banyak dokumen yang Anda perlu lihat untuk menjawab ini?**

Disini serp B lebih bagus karena behaviour penggunanya

Jadi kalau memang orang butuh banyak dokumen, orang lebih pilih SERP B.

## SERP A (Ranking A)

D8 D7 D91 D92 D93 D94 D95 D96 D97 D98

## SERP B (Ranking B)

D5 D7 D8 D1 D3 D4 D6 D2 D83 D84

**Jika metric score antara 0 dan 1, berapa Anda berikan score untuk 2 ranking di atas?**

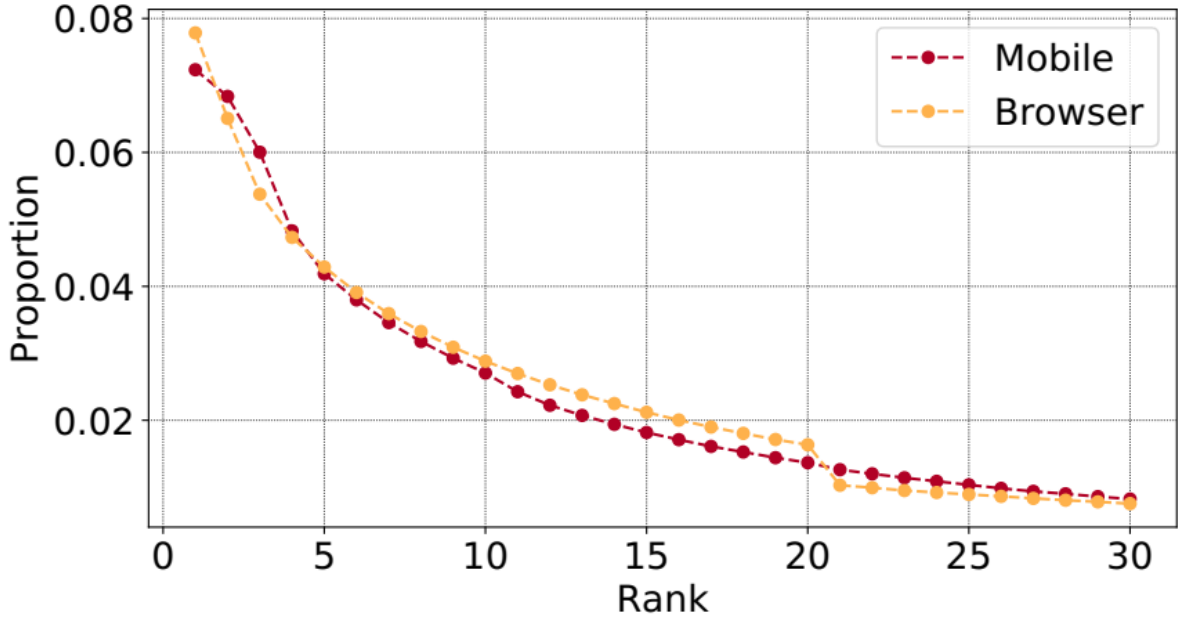
# Jadi ...

- Nilai yang dihasilkan sebuah metric perlu bergantung kepada **user behavior**. highlight this

- Dan **user behavior** dikendalikan oleh "search goal", yaitu "berapa banyak dokumen relevan yang dibutuhkan".

- User behavior --> **User model**
  - How users interact with SERP
  - Bagaimana pola user **"melihat"** dokumen-dokumen yang ada di SERP?

# Bagaimana User Inspeksi SERP?

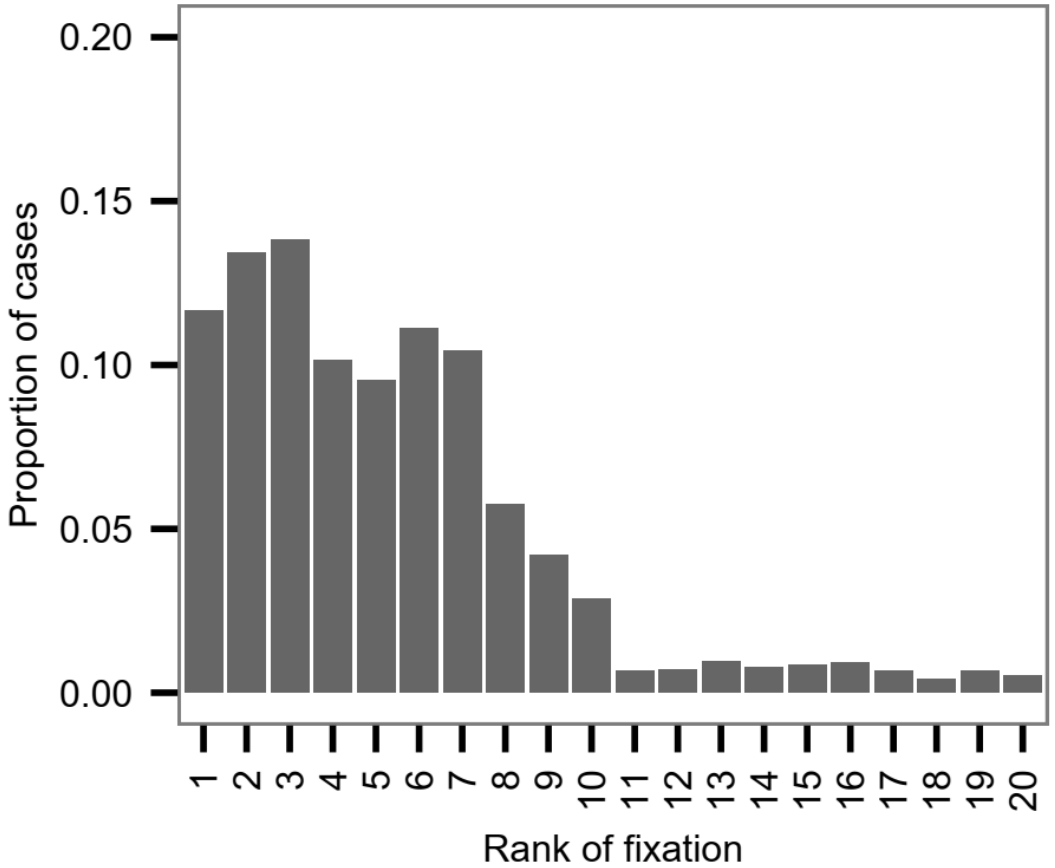



Hasil eksperimen dengan signal impression pada data SEEK.com job search engine.

**Wicaksono & Moffat, CIKM 2018**

Hasil eksperimen dengan alat **eye tracker**.

**Paul Thomas, ADCS 2013**



# Metric Based on User Model

Misal, ranking/SERP = $\vec{r} = [r_1, r_2, r_3, r_4, r_5, \dots]$

dimana $r_i$ = **1** jika relevan dan $r_i$ = **0** jika tidak.

Bentuk umum metric **M**:

**In practice: biasanya Top-1000**

$$M@K(\vec{r}) = \sum_{i=1}^{K} D(i).r_i$$

D1 harusnya lebih gede dari D2 karena ranked

**K: top-K results**

**D(i) decreases with rank i**

Sebuah **discount function** yang proporsional terhadap "**probabilitas user inspeksi posisi rank i**".

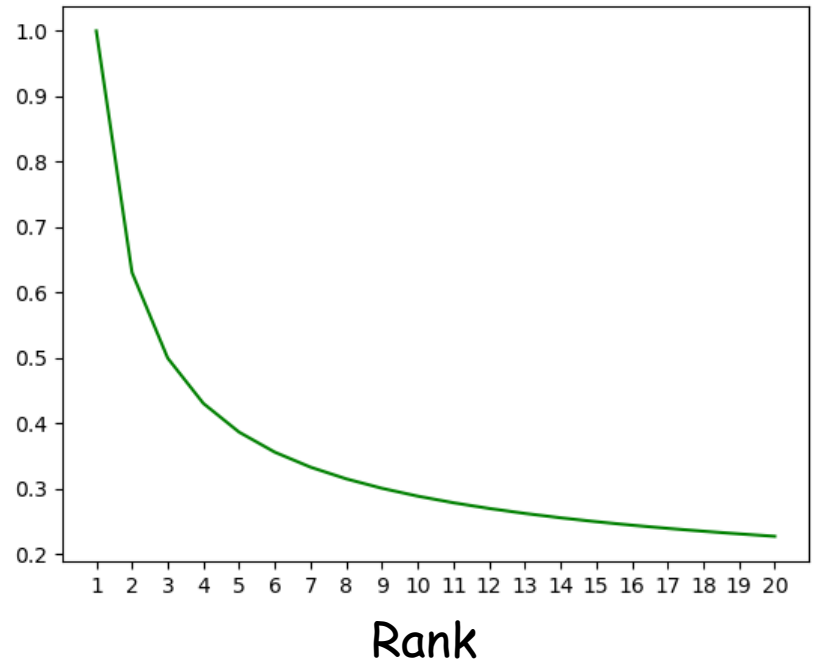
# Discounted Cumulative Gain (DCG)

Jarvelin & Kekalainen 2002

$$DCG@K(\vec{r}) = \sum_{i=1}^{K} \frac{1}{\log_2(i+1)} . r_i$$

Probabilitas user inspeksi rank i

$1/\log_2(i+1)$




Score bisa bernilai > 1.

**Interpretasi score**: ***expected total volume of relevance (expected total gain)*** yang dikumpulkan seorang user.

jadi kalo ada 2 model dimana 2-2nya itu cuman dapetin 5 query relevan dari 20 query yang dijalankan

jadi precisionnya 2-2nya itu 5/20. Nah kalau kita pakai discount, apabila model A itu query yang awal2 yang di retrieve misalnya (1, 2, 3, 4, 5) vs model B yang retrieve dokumen relevan yang D15, D16, D17, D18, D19. Nah keliatan kalau model A lebih bagus.

* Ini sebenarnya bukan versi DCG asli yang diusulkan oleh Jarven & Kekalainen.
Ini adalah versi modifikasi yang diusulkan oleh peneliti dari **Microsoft**.

# Normalized DCG (NDCG)

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

agar upperbound scoreenya itu 1 or known as normalisasi

DCG@K dibagi dengan DCG@K **ketika "ranking ideal"**

Contoh:

sebuah ranking **r = [0, 1, 0, 1, 1]**,

dengan asumsi hanya ada 3 relevant documents di koleksi.

**$r_{ideal}$ = [1, 1, 1, 0, 0]**

R ideal itu maksudnya semua dokumennya diatas semua (atau discounted pricenya yang terbesar dari query yang ada)

$$DCG@5(\boldsymbol{r}) = \frac{0}{\log_2(2)} + \frac{1}{\log_2(3)} + \frac{0}{\log_2(4)} + \frac{1}{\log_2(5)} + \frac{1}{\log_2(6)} = 1.45$$

$$IDCG@5(\boldsymbol{r}) = \frac{1}{\log_2(2)} + \frac{1}{\log_2(3)} + \frac{1}{\log_2(4)} + \frac{0}{\log_2(5)} + \frac{0}{\log_2(6)} = 2.13$$

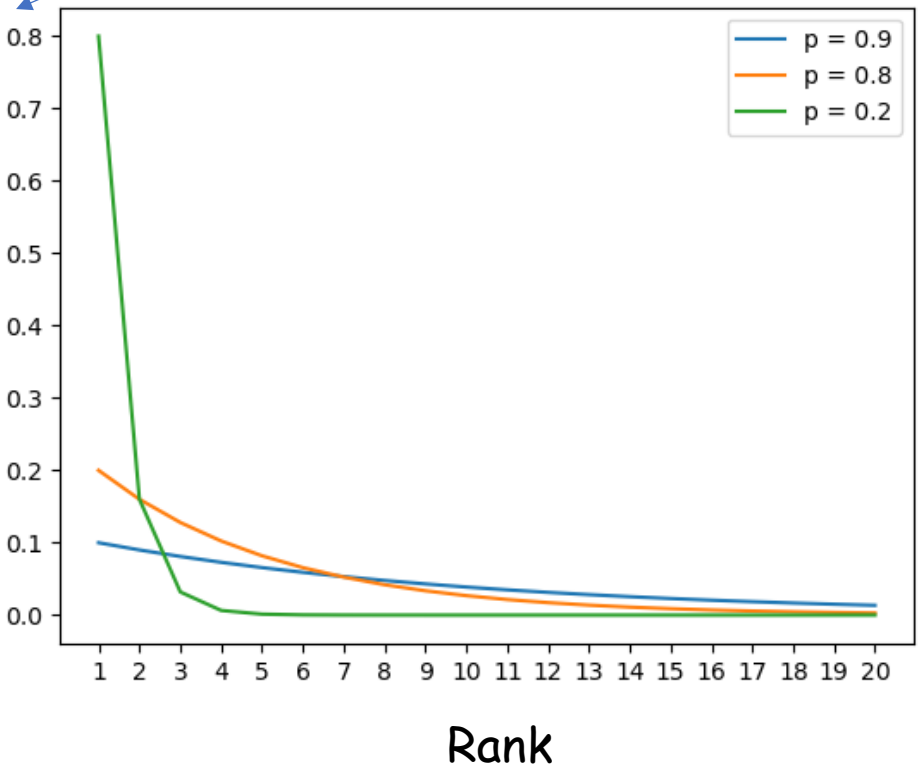
$$NDCG@5(\boldsymbol{r}) = \frac{1.45}{2.13} = 0.68$$

# Rank Biased Precision

## Moffat & Zobel, ACM TOIS 2008

$$RBP@K(\vec{r}; p) = \sum_{i=1}^{K} (1-p).p^{(i-1)}.r_i$$

Fraction of rank i inspected by a population of users.




Score antara 0 dan 1.

**Interpretasi score**: ***expected rate of gain*** yang dikumpulkan seorang user.

**p** adalah parameter yang merepresentasikan **"tingkat kesabaran user"** saat inspeksi SERP. Biasanya pakai **p = 0.8**.

p = 0 cuman lihat rank 1 (paling atas aja)

**p** tinggi: user dengan senang hati inspeksi sampai dokumen ranking bawah
**p** rendah: user hanya mau inspeksi dokumen di ranking atas saja

# Precision@K

Artinya, setiap posisi rank **i** punya **probabilitas yang sama** untuk diinspeksi oleh user, yaitu **1/K**.

$$Prec@K(\vec{r}) = \sum_{i=1}^{K} \frac{1}{K} . r_i$$

kalau ini kayak Discounted Price cuman nilainya itu 1/<TOP K>

Score antara 0 dan 1.

**Interpretasi score**: ***expected rate of gain*** yang dikumpulkan seorang user.

# Average Precision

Score antara 0 dan 1.

**Interpretasi score**: ***expected rate of gain*** yang dikumpulkan seorang user.

$$AP@K(\vec{r}) = \sum_{i=1}^{K} \frac{Prec@i(\vec{r})}{R} . r_i$$

jadi precision i (R) itu precision di dalem SERP kita

misalnya SERP = [1, 0, 1, 0, 1]

jadi bakal kayak

Precision index 1 -> 1/1
Precision index 2 -> 1/2 (karena 0 gk relevan)
Precision index 3 -> 2/3
prec index 4 -> 2/4
Precision index 5 -> 3/5

So the result will be:

1/3 * 1 + 0.5/3 * 0 + 2/9 * 1 + 0.5/3 * 0 + 3/15 * 1

R = nilai yang ada di SERP/retrieval relevance kita (bukan di QRELS)

R disini nilainya itu 3 karena ada 3 dokumen yang relevan di koleksi kita.

**R** = banyaknya dokumen relevan di koleksi
**R** biasanya jarang diketahui. Mengapa?

actually R is based on this.

Terkadang **R** diaproksimasi dengan $R = \sum_{i=1}^{K} r_i$

# Latihan

$\vec{r} = [1,1,0,0,1,0,1,0,0,1]$

Hitunglah score (akan lebih mudah jika membuat program)

- RBP@10, dengan p = 0.8
- DCG@5
- DCG@10
- Prec@5
- AP@5

Untuk AP, gunakan aproksimasi R:

$$R = \sum_{i=1}^{K} r_i$$

Source: CS6200 Information Retrieval, Northeastern University

# Statistical Significance

- Information Retrieval and other experimental sciences aim to compare different systems and determine if their outcomes are "really" distinct.

- For example, "Does BM25 really outperform TFIDF?"

- Statistical hypothesis testing is a tool to help us making justified conclusions from our experimental results.

Source: CS6200 Information Retrieval, Northeastern University

# Hypothesis Testing

- The test allows us to choose between the null hypothesis and an alternative hypothesis.
  - Null hypothesis: Model A is not different from Model B
  - Alternative hypothesis: Model A is different from B (the change improved performance)

- A hypothesis test doesn't directly confirm the alternative hypothesis. It calculates the likelihood that the observed data could have occurred by chance, even if the null hypothesis is true.

Source: CS6200 Information Retrieval, Northeastern University

# Test Steps

jangan banyak-banyak biar gak ambigu

- Prepare your experiment carefully, with **only one difference between the two systems**: the change whose effect you wish to measure. Choose a **significance level** $\alpha$, used to make your decision.

biasanya alfa itu 0.05

- Run each system many times (e.g. on many different queries, 25 queries or 50 queries, …), evaluating each run with a metric (e.g. AP, RBP, DCG, or other metrics).

biar bisa confident dengan hipotesis value kita (p -value)

- Calculate a test statistic for each system based on the distributions of evaluation metrics.

Source: CS6200 Information Retrieval, Northeastern University

# Test Steps

- Use a statistical significance test to compare the test statistics (one for each system). This will give you a **p-value**: **the probability of the null hypothesis producing a difference at least this large**.
- If the **p-value** is less than $\alpha$, reject the null hypothesis.

kalau p value itu lebih gede, kita gak punya bukti yang kuat

kalau p value itu lebih kecil dari alpha, maka kita bisa reject null hypotesis dan kita ambil kesimpulan bahwa kesimpulan awal benar

- The probability that you will correctly reject the null hypothesis using a particular statistical test is known as its **power**.

p value = proability value:
- dengan mengasumsikan null hypotesisnya benar, kita bisa pilih untuk mengambil alternative hypthosesis atau engga

# Paired t-test

H0: Score Model A dan Model B tidak berbeda
H1: Score Model A dan Model B berbeda

Lakukan eksperimen terhadap model A dan model B terhadap **b** buah *queries*.

Kita dapat menghitung **p-value** untuk mengetahui apakah metric score yang dihasilkan model A dan model B memang berbeda --> paired t-test

| Query | Score A | Score B | $D$ |
|---|---|---|---|
| 1 | $M_1^A$ | $M_1^B$ | $M_1^A - M_1^B$ |
| 2 | $M_2^A$ | $M_2^B$ | $M_2^A - M_2^B$ |
| … | … | … | … |
| b | $M_b^A$ | $M_b^B$ | $M_b^A - M_b^B$ |

**p-value = 2 x P(T > |t|)**

$$t = \sqrt{b}\frac{\overline{D}}{SD(D)}$$

dimana,

$$SD(D) = \sqrt{\frac{(D_1 - \overline{D})^2 + \cdots + (D_b - \overline{D})^2}{b-1}}$$

# Paired t-test

H0: Score Model A dan Model B tidak berbeda
H1: Score Model A dan Model B berbeda

Lakukan eksperimen terhadap ( ) buah *queries*.

Nilai t mengikuti distribusi t-student dengan degree of freedom = b - 1

Kita dapat menghitung **p-value** ... ic score yang dihasilkan model A dan model B ... ang berbeda --> paired t-test

| Query | Score A | Score B | $D$ |
|---|---|---|---|
| 1 | $M_1^A$ | $M_1^B$ | $M_1^A - M_1^B$ |
| 2 | $M_2^A$ | $M_2^B$ | $M_2^A - M_2^B$ |
| … | … | … | … |
| b | $M_b^A$ | $M_b^B$ | $M_b^A - M_b^B$ |

$$t = \sqrt{b}\frac{\overline{D}}{SD(D)}$$

dimana,

$$SD(D) = \sqrt{\frac{(D_1 - \overline{D})^2 + \cdots + (D_b - \overline{D})^2}{b-1}}$$

**p-value = 2 x P(T > |t|)**

**Biasanya H0 ditolak jika p-value < 0.05**

# Contoh Praktikal dengan Scipy

- Misal, kira run untuk *12 queries*
- Untuk setiap run, nilai metrik (misal *AP*) dihitung.

```
AP_A = [32.3, 20.3, 31.4, 25.7, 28.4, 27.3, 29.3, 30.1, 25.5, 28.7, 29.1, 24.8]

AP_B = [32.0, 20.4, 31.2, 25.0, 27.9, 26.9, 29.1, 30.0, 24.4, 28.2, 28.6, 24.6]

from scipy import stats
print(stats.ttest_rel(AP_A, AP_B))
# Ttest_relResult(statistic=4.244464615962889, pvalue=0.0013784945927875687)
```

Karena p < 0.05 (significance value), ada evidence yang kuat bahwa model A brbeda dengan model B

Buat yang lebih baik itu lihat dari rata-rata.

kalo misal banyak evidence nih kalau AP_A lebih bagus dibadingin AP_B, maka p value nya tentu aja bakal kecil banget. CUman kalau memang cukup kompetitif. Maka p1 nya juga akan lebih besar,

# Contoh Praktikal dengan Scipy

- Misal, kira run untuk *12 queries*
- Untuk setiap run, nilai metrik (misal *AP*)

```
AP_A = [32.3, 20.3, 31.4, 25.7, 28.4, 27.3, 29.3, 30.1, 2

AP_B = [32.0, 20.4, 31.2, 25.0, 27.9, 26.9, 29.1, 30.0, 24.

from scipy import stats
print(stats.ttest_rel(AP_A, AP_B))
# Ttest_relResult(statistic=4.244464615962889, pvalue=0.0013784945927875687)
```

p-value < 0.05; ada evidence model B lebih baik dari model A

# Lebih detail terkait p-value?

- P-value, or probability value = how likely it is that your data could have occurred if the null hypothesis were true.

- If your *p*-value is 0.05, that means that 5% of the time you would see a test statistic at least as extreme as the one you found if the null hypothesis was true.

- The smaller the *p*-value, the more likely you are to reject the null hypothesis.
  - Jika p-value kecil, artinya "jangan-jangan" H0 telah salah, karena saya telah melihat data/observasi yang probabilitas kemunculannya kecil.

Source: CS6200 Information Retrieval, Northeastern University

# Error Types

- Hypothesis testing involves balancing between two types of errors:

- **Type I Errors**, or **false positives**, occur when the null hypothesis is true, but you reject it.
  - The probability of type I error = alpha = significance level

- **Type II Errors**, or **false negatives**, occur when the null hypothesis is false, but you don't reject it.
  - The probability of type II error = beta = 1 - power