

Ujian Tengah Semester

Sistem Paralel dan Terdistribusi A

Laporan Pub-Sub Log Aggregator dengan Idempotent Consumer dan Deduplication



Disusun Oleh :

Zakaria Fattawari 11231092

24 Oktober 2025

BAGIAN TEORI

1. Jelaskan karakteristik utama sistem terdistribusi dan *trade-off* yang umum pada desain Pub-Sub log aggregator!
2. Bandingkan arsitektur *client-server* vs *publish-subscribe* untuk aggregator. Kapan memilih Pub-Sub? Berikan alasan teknis!
3. Uraikan *at-least-once* vs *exactly-once delivery semantics*. Mengapa idempotent consumer krusial di *presence of retries*?
4. Rancang skema penamaan untuk topic dan event_id (unik, *collision-resistant*). Jelaskan dampaknya terhadap dedup.
5. Bahas *ordering*: kapan total *ordering* tidak diperlukan? Usulkan pendekatan praktis (mis. *event timestamp* + *monotonic counter*) dan batasannya!
6. Identifikasi *failure modes* (duplikasi, *out-of-order*, *crash*). Jelaskan strategi mitigasi (*retry*, *backoff*, *durable dedup store*)!
7. Definisikan *eventual consistency* pada aggregator; jelaskan bagaimana idempotency + dedup membantu mencapai konsistensi.
8. Rumuskan metrik evaluasi sistem (*throughput*, *latency*, *duplicate rate*) dan kaitkan ke keputusan desain.

Jawaban:

1. Sistem terdistribusi dicirikan oleh *openness*, *resource sharing*, *concurrency*, *scalability*, *transparency*, dan *fault tolerance* (Tanenbaum & Van Steen, 2023, hlm. 18–25). Pada pola *publish-subscribe* berbasis *message-queuing*, produsen dan konsumen *decoupled* secara temporal: pesan disimpan persisten di *queue* sehingga pengirim/penerima tak perlu aktif bersamaan (Tanenbaum & Van Steen, 2023, hlm. 72–79; Tanenbaum & Van Steen, 2023, hlm. 198–205). Ini meningkatkan elastisitas, isolasi *back-pressure*, serta *availability*, tetapi ada *trade-off*: latensi tambahan untuk durabilitas, ketidakpastian *delivery time*, dan kompleksitas *routing/matching* saat ekspresivitas *subscription* meningkat (Tanenbaum & Van Steen, 2023, hlm. 206–220). Untuk log aggregator, Pub-Sub memberi *fan-out* efisien dan *buffering* saat lonjakan trafik, namun menuntut konsumen idempoten serta *deduplication* karena *retries* memicu duplikasi (Tanenbaum & Van Steen, 2023, hlm. 408–420). Selain itu, *ordering* global sering tidak realistis; arsitektur perlu toleran *out-of-order* dan memilih *ordering* lebih lemah (mis. FIFO/kausal) jika memadai (Tanenbaum & Van Steen, 2023, hlm. 264–280).

2. Arsitektur *client-server* cocok untuk *request-response* sinkron dengan *endpoint identity* yang jelas dan jalur kontrol sempit; namun ia *temporally coupled*—klien dan server harus aktif bersamaan (Tanenbaum & Van Steen, 2023, hlm. 72–85). *Publish-subscribe* memadankan *subscription* terhadap *notification* berbasis topik/atribut dan memberi *decoupling in time* berkat *asynchronous queues* (Tanenbaum & Van Steen, 2023, hlm. 198–205). Pub-Sub unggul saat *fan-out* besar, *dynamic membership*, dan kebutuhan *loose coupling*—tetapi menambah kompleksitas *matching*, kebutuhan *broker/queue* yang andal, serta kemungkinan *delivery latency* lebih tinggi (Tanenbaum & Van Steen, 2023, hlm. 206–220). Pilih Pub-Sub untuk *event dissemination* skala besar/log aggregation; pilih *client-server* untuk RPC kontrol yang sempit atau ketika *endpoint identity* dan *low latency* sangat krusial (Tanenbaum & Van Steen, 2023, hlm. 72–79; Tanenbaum & Van Steen, 2023, hlm. 206–214).
3. *At-least-once* menjamin eksekusi minimal sekali namun membuka peluang duplikasi saat *retries*; *at-most-once* menghindari duplikasi tetapi berisiko kehilangan pesan; *exactly-once* sulit dicapai secara end-to-end di hadapan *crash* dan ketidakpastian *acknowledgement* (Tanenbaum & Van Steen, 2023, hlm. 206–235). Karena itu, konsumen harus idempoten, misalnya dengan kunci (`topic`, `event_id`) dan *dedup store* untuk menolak *replay* yang sama; bila *payload* dapat berubah, tambahkan *content hash* untuk mendeteksi konflik (Tanenbaum & Van Steen, 2023, hlm. 236–250; Tanenbaum & Van Steen, 2023, hlm. 408–420). Pendekatan ini memberikan *exactly-once effect* di tingkat konsumen walau jaringan/messaging hanya menawarkan jaminan yang lebih lemah (Tanenbaum & Van Steen, 2023, hlm. 206–214; Tanenbaum & Van Steen, 2023, hlm. 408–412).
4. *Naming* memisahkan identifier dari address, mendukung *location independence* dan perubahan lokasi/implementasi tanpa mengubah identitas (Tanenbaum & Van Steen, 2023, hlm. 342–352). Gunakan `event_id` berbasis flat naming (UUID/ULID) atau self-certifying names (berbasis *hash*/kunci publik) guna menurunkan probabilitas tabrakan dan mendukung verifikasi (Tanenbaum & Van Steen, 2023, hlm. 360–372). `topic` sebaiknya structured (misalnya, `logs.app.component`) untuk pengelompokan dan *resolution* yang konsisten (Tanenbaum & Van Steen, 2023, hlm.

352–360). Kombinasi (`topic`, `event_id`) memudahkan dedup; jika terjadi tabrakan *ID*, *content hash* menjadi *tie-breaker* untuk membedakan duplikasi sejati dari konflik (Tanenbaum & Van Steen, 2023, hlm. 372–385).

5. Banyak aliran log tidak memerlukan total order; cukup causal/FIFO ketika tidak ada dependensi kuat antar-event (Tanenbaum & Van Steen, 2023, hlm. 264–273). *Lamport clocks* menjaga urutan *happens-before* namun tidak membuktikan kausalitas; *vector clocks* memungkinkan deteksi potensi kausalitas dan *causally ordered delivery* dengan biaya metadata lebih besar (Tanenbaum & Van Steen, 2023, hlm. 274–300). Praktik pragmatis untuk aggregator: gunakan (`event_timestamp`, *monotonic per-producer counter*) sebagai *tie-breaker*, izinkan *late arrival* dan lakukan *reordering* lokal sebatas *window* tertentu (Tanenbaum & Van Steen, 2023, hlm. 300–320). Batasan: *clock skew*, *drift*, dan overhead metadata untuk *vector clocks* (Tanenbaum & Van Steen, 2023, hlm. 280–320).
6. Antrian persisten memutus *temporal coupling* tetapi hanya menjamin enqueue, bukan *delivery time*—ini mendorong retry dan potensi duplikasi (Tanenbaum & Van Steen, 2023, hlm. 206–220). *Out-of-order* lazim pada multi-produser/partisi dan variabilitas jaringan; *crash* menimbulkan ketidakpastian *ack* (Tanenbaum & Van Steen, 2023, hlm. 220–235). Mitigasi: exponential backoff, idempotent consumer dengan durable dedup store (tahan gangguan), serta *write-ahead record/insert* sebelum *side-effects* agar dapat *replay* aman (Tanenbaum & Van Steen, 2023, hlm. 236–250; Tanenbaum & Van Steen, 2023, hlm. 408–420). Untuk pesan bermasalah, pisahkan ke *dead-letter/poison queue* guna analisis tanpa menghambat aliran utama (Tanenbaum & Van Steen, 2023, hlm. 214–220; Tanenbaum & Van Steen, 2023, hlm. 408–412).
7. *Eventual consistency* menyatakan bahwa—tanpa konflik tulis—semua replika akhirnya konvergen seiring propagasi update (Tanenbaum & Van Steen, 2023, hlm. 408–430). Idempotency memastikan *replay* tidak mengubah hasil; deduplication mencegah *double-apply* akibat *retries/redelivery* (Tanenbaum & Van Steen, 2023, hlm. 430–452). Ketika konflik tak terhindarkan, terapkan resolusi sederhana (mis. last-writer-wins berbasis jam yang disepakati) atau skema berbasis *causal ordering* untuk menjaga koherensi praktis (Tanenbaum & Van Steen, 2023, hlm. 452–480).

8. *Metrik*:

- a. *Throughput* (events/s),
- b. *End-to-end latency*,
- c. *Duplicate rate*,
- d. *Error rate*, dan
- e. *Lag* (selisih `event.timestamp` \rightarrow `commit`) (Tanenbaum & Van Steen, 2023, hlm. 72–79; Tanenbaum & Van Steen, 2023, hlm. 198–205).

Keputusan desain:

- a. Kunci idempoten (`topic`, `event_id`) + content hash untuk *conflict detection* (Tanenbaum & Van Steen, 2023, hlm. 372–385),
- b. Durable dedup store untuk *exactly-once effect* di konsumen (Tanenbaum & Van Steen, 2023, hlm. 408–420),
- c. *Ordering* praktis via *timestamp* + *counter* (tanpa *global total order*) (Tanenbaum & Van Steen, 2023, hlm. 264–300), dan
- d. Retry + backoff pada *message-queuing* (Tanenbaum & Van Steen, 2023, hlm. 206–220).

Arsitektur:

Publisher to Broker/Exchange to Queue (*durable*) to Idempotent Aggregator (*dedup store*) to Sink/Analytics; *metrics endpoint* untuk observabilitas (Tanenbaum & Van Steen, 2023, hlm. 72–85; Tanenbaum & Van Steen, 2023, hlm. 408–430).

Tanenbaum, A. S., & Van Steen, M. (2023). *Distributed systems* (Edisi ke-4). Addison-Wesley.

DOKUMENTASI

Build and Run Docker File

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> docker build -t uts-aggregator .
[+] Building 0.3s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 410B
=> [internal] load metadata for docker.io/library/python:3.11-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/6] FROM docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> => resolve docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> [internal] load build context
=> => transferring context: 682B
=> CACHED [2/6] WORKDIR /app
=> CACHED [3/6] RUN adduser --disabled-password --gecos '' appuser && mkdir -p /app/data && chown -R appuser:appuser /app
=> CACHED [4/6] COPY requirements.txt .
=> CACHED [5/6] RUN pip install --no-cache-dir -r requirements.txt
=> CACHED [6/6] COPY src/ ./src/
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:5364ce57ad33222ef528e50ead2cf29788ee60b5766622ef76651c78a5879293
=> => exporting config sha256:ce84dc1666a8c0f7c04154379dd0f036d14a6e1bed824b255632099837d2dafa
=> => exporting attestation manifest sha256:9c54b8784f9dc4ef5c86473c1505c8f12407ef10ab32985b338147d00c8eaa54
=> => exporting manifest list sha256:e22fcc58c3debc7ba8d3ef9fbb33e90f08cec131e5c8b9309b5c5932f54c9f22
=> => naming to docker-desktop://dashboard/build/desktop-linux/desktop-linux/vebuzyulfi2cy3do7ofmu42a3 (ctrl + click)
=> => unpacking to docker-desktop://dashboard/build/desktop-linux/desktop-linux/vebuzyulfi2cy3do7ofmu42a3
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/vebuzyulfi2cy3do7ofmu42a3
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> docker run -p 8080:8080 uts-aggregator
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
```

Docker Compose

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> docker compose up --build
time="2025-10-23T23:40:36+08:00" level=warning msg="E:\\Semester 5\\Parallel and Distributed System\\Ujian Tengah Semester\\uts-pubsub-aggregator\\docker-compose.yml: the attribute `version` is obsolete
, it will be ignored, please remove it to avoid potential confusion"
[+] Building 0.4s (13/13) FINISHED
=> [internal] load local bake definitions                                0.0s
=> => reading from stdin 6848                                           0.0s
=> [internal] load build definition from Dockerfile                     0.0s
=> => transferring dockerfile: 4188                                      0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim      0.0s
=> [internal] load .dockerignore                                         0.0s
=> => transferring context: 28                                           0.0s
=> [1/6] FROM docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0 0.0s
=> => resolve docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0 0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 682B                                          0.0s
=> CACHED [2/6] WORKDIR /app                                             0.0s
=> CACHED [3/6] RUN adduser --disabled-password --gecos '' appuser && mkdir -p /app/data && chown -R appuser:appuser /app 0.0s
=> CACHED [4/6] COPY requirements.txt .                                  0.0s
=> CACHED [5/6] RUN pip install --no-cache-dir -r requirements.txt      0.0s
=> CACHED [6/6] COPY src/ ./src/                                         0.0s
=> => exporting to image                                                 0.1s
=> => exporting layers                                                  0.0s
=> => exporting manifest sha256:d4edf6c9d637c024e5e25a08e5eaf4cc1ab616a7a8e2341eb8bad7b13d618ba5 0.0s
=> => exporting config sha256:13145ac720d81b0d8706ba2e72b01aa4029c83524672b0e3c7b796e75c1ec10 0.0s
=> => exporting attestation manifest sha256:2110a36f4d36e0db749df4f7799a708594f6f3d664fc6beca18eacde17f52680 0.0s
=> => exporting manifest list sha256:c8c4ebafb9ac6d2e1f2379e54fe30e2488848fd6265147636b4055687ed3360a 0.0s
=> => naming to docker.io/library/uts-pubsub-aggregator-aggregator:latest 0.0s
=> => unpacking to docker.io/library/uts-pubsub-aggregator-aggregator:latest 0.0s
=> resolving provenance for metadata file                                0.0s
[+] Running 4/4
✔ uts-pubsub-aggregator-aggregator Built 0.0s
✔ Network uts-pubsub-aggregator_default Created 0.0s
✔ Container uts-pubsub-aggregator-aggregator-1 Created 0.1s
✔ Container uts-pubsub-aggregator-publisher-1 Created 0.1s
Attaching to aggregator-1, publisher-1
```

Lihat OpenAPI Docs

The screenshot shows the FastAPI OpenAPI documentation interface. At the top, the browser address bar displays 'localhost:8080/docs'. The FastAPI logo is visible, along with version indicators '0.1.0' and 'OAS 3.1'. The interface is divided into two main sections: 'default' and 'Schemas'.

The 'default' section lists the following endpoints:

- POST** `/publish` Publish
- GET** `/events` Get Events
- GET** `/stats` Get Stats
- GET** `/` Root

The 'Schemas' section lists the following data models:

- Event** Expand all object
- HTTPValidationError** Expand all object
- PublishRequest** Expand all object
- ValidationError** Expand all object

POST /publish

Responses


Curl

```
curl -X 'POST' \
'http://localhost:8080/publish' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "events": [
    {
      "topic": "logs.appA",
      "event_id": "event-0002",
      "timestamp": "2025-01-01T01:00:00Z",
      "source": "cli",
      "payload": {
        "level": "info",
        "msg": "first log"
      }
    },
    {
      "topic": "logs.appA",
      "event_id": "event-0002",
      "timestamp": "2025-01-01T01:00:00Z",
      "source": "cli",
      "payload": {
        "level": "info",
        "msg": "duplicate log"
      }
    },
    {
      "topic": "logs.appA",
      "event_id": "event-0003",
      "timestamp": "2025-01-01T02:00:00Z",
      "source": "cli",
      "payload": {
        "level": "warn",
        "msg": "another event"
      }
    }
  ]
}'
```

Request URL

```
http://localhost:8080/publish
```

Server response

Code	Details
200	<div>Response body<pre>{ "enqueued": 3, "queue_size": 0 }</pre><div> Download</div></div> <div>Response headers<pre>content-length: 20 content-type: application/json date: Thu, 23 Oct 2025 15:41:59 GMT server: uvicorn</pre></div>

Responses

Code	Description	Links
------	-------------	-------

GET /events

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/events?topic=logs.app&limit=1000' \
  -H 'accept: application/json'
```

Request URL

http://localhost:8080/events?topic=logs.app&limit=1000

Server response

Code

Details

200

Response body

```
[
  {
    "topic": "logs.appA",
    "event_id": "event-0002",
    "timestamp": "2025-01-01T01:00:00+00:00",
    "source": "cli",
    "payload": {
      "level": "info",
      "msg": "first log"
    },
    "processed_at": 1761234119.5212462
  },
  {
    "topic": "logs.appA",
    "event_id": "event-0003",
    "timestamp": "2025-01-01T02:00:00+00:00",
    "source": "cli",
    "payload": {
      "level": "warn",
      "msg": "another event"
    },
    "processed_at": 1761234119.5215724
  }
]
```

Download

Response headers

```
content-length: 365
content-type: application/json
date: Thu, 23 Oct 2025 15:43:04 GMT
server: uvicorn
```

Responses

Code

Description

Links

9

GET /stats

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/stats' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/stats
```

Server response

Code

Details

200

Response body

```
{
  "received": 3,
  "unique_processed": 2,
  "duplicate_dropped": 1,
  "topics": [
    "logs.appA"
  ],
  "uptime_seconds": 211.175
}
```

Download

Response headers

```
content-length: 185
content-type: application/json
date: Thu, 23 Oct 2025 15:43:37 GMT
server: uvicorn
```

Responses

Code

Description

Links

Test Duplication - POST /publish

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> $body = @({
>>   events = @(
>>     @{
>>       topic="logs.appA"; event_id="event-0002"; timestamp="2025-01-01T01:00:00Z"; source="cli";
>>       payload=@{ level="info"; msg="first log" }
>>     },
>>     @{
>>       topic="logs.appA"; event_id="event-0002"; timestamp="2025-01-01T01:00:00Z"; source="cli";
>>       payload=@{ level="info"; msg="duplicate log" }
>>     },
>>     @{
>>       topic="logs.appA"; event_id="event-0003"; timestamp="2025-01-01T02:00:00Z"; source="cli";
>>       payload=@{ level="warn"; msg="another event" }
>>     }
>>   )
>> } | ConvertTo-Json -Depth 6
>> Invoke-RestMethod -Uri "http://localhost:8080/publish" -Method POST -ContentType "application/json" -Body $body

enqueued queue_size
-----
3          0
```

Test Duplication - GET /stats

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> Invoke-RestMethod -Uri "http://localhost:8080/stats"

received      : 6
unique_processed : 2
duplicate_dropped : 4
topics        : {logs.appA}
uptime_seconds : 263.642
```

Test Duplication - GET /events topic=logs.appA

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> Invoke-RestMethod -Uri "http://localhost:8080/events?topic=logs.appA"

topic      : logs.appA
event_id   : event-0002
timestamp  : 2025-01-01T01:00:00+00:00
source     : cli
payload    : @{"level=info; msg=first log"}
processed_at : 1761234119.5212462

topic      : logs.appA
event_id   : event-0003
timestamp  : 2025-01-01T02:00:00+00:00
source     : cli
payload    : @{"level=warn; msg=another event"}
processed_at : 1761234119.5215724
```

Idempotency

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> $e = @{ topic="persist"; event_id="persist-1"; timestamp="2025-01-01T00:00:00Z"; source="cli"; payload=@{ } }
>> $body = @([events:@($e)] | ConvertTo-Json -Depth 5)
>> Invoke-RestMethod -Uri "http://localhost:8080/publish" -Method POST -ContentType "application/json" -Body $body

enqueued queue_size
-----
1                0
```

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> Invoke-RestMethod -Uri "http://localhost:8080/events?topic=persist"

topic      : persist
event_id   : persist-1
timestamp  : 2025-01-01T00:00:00+00:00
source     : cli
payload    :
processed_at : 1761234384.6498368
```

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> docker stop $(docker ps -q --filter ancestor=uts-aggregator)
506dc2c3f76b
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> docker run -p 8080:8080 -v aggdata:/app/data uts-aggregator
INFO:      Started server process [1]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
```

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> $e = @{ topic="persist"; event_id="persist-1"; timestamp="2025-01-01T00:00:00Z"; source="cli"; payload=@{ } }
>> $body = @([events:@($e)] | ConvertTo-Json -Depth 5)
>> Invoke-RestMethod -Uri "http://localhost:8080/publish" -Method POST -ContentType "application/json" -Body $body

enqueued queue_size
-----
1                0
```

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> Invoke-RestMethod -Uri "http://localhost:8080/stats"

received      : 1
unique_processed : 1
duplicate_dropped : 0
topics        : {persist}
uptime_seconds : 34.045
```

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> Invoke-RestMethod -Uri "http://localhost:8080/events?topic=persist"

topic      : persist
event_id   : persist-1
timestamp  : 2025-01-01T00:00:00+00:00
source     : cli
payload    :
processed_at : 1761234378.4918468
```

Schema Validation

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> $bad = @{ events=@(
>>   @{ topic="bad"; event_id="id-1"; timestamp="2025-01-01T00:00:00Z"; source="cli"; payload=@{} }
>> )} | ConvertTo-Json -Depth 5
>>
>> Invoke-RestMethod -Uri "http://localhost:8080/publish" -Method POST -ContentType "application/json" -Body $bad

enqueued queue_size
-----
1             0
```


Mini Stress: 5.000 events with 20% duplicate

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> $base = @{ topic="stress"; source="tester"; timestamp=(Get-Date).ToUniversalTime().ToString("o") }
>> $events = @()
>> 0..3999 | ForEach-Object { $events += @{ topic=$base.topic; source=$base.source; timestamp=$base.timestamp; event_id="$$_"; payload=@{i=$_} } }
>> 0..999 | ForEach-Object { $events += @{ topic=$base.topic; source=$base.source; timestamp=$base.timestamp; event_id="$$_"; payload=@{i=$_} } }
>>
>> $body = @($events) | ConvertTo-Json -Depth 6
>> Invoke-RestMethod -Uri "http://localhost:8080/publish" -Method POST -ContentType "application/json" -Body $body

enqueued queue_size
-----
5000 0
```

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> Invoke-RestMethod -Uri "http://localhost:8080/stats"

received      : 5002
unique_processed : 4002
duplicate_dropped : 1000
topics        : {bad, persist, stress}
uptime_seconds : 109.299
```

Unit Test

```
PS E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator> pytest -v
===== test session starts =====
platform win32 -- Python 3.11.6, pytest-8.3.3, pluggy-1.6.0 -- C:\Users\USER\AppData\Local\Programs\Python\Python311\python.exe
cachedir: .pytest cache
rootdir: E:\Semester 5\Parallel and Distributed System\Ujian Tengah Semester\uts-pubsub-aggregator
plugins: anyio-4.11.0, locust-2.41.6
collected 5 items

tests/test_dedup.py::test_dedup_unique_once PASSED [ 20%]
tests/test_persistence.py::test_persistence_survives_restart_sim PASSED [ 40%]
tests/test_schema.py::test_schema_validation PASSED [ 60%]
tests/test_stats_events.py::test_stats_and_events_consistency PASSED [ 80%]
tests/test_stress.py::test_stress_5k_events PASSED [100%]

===== 5 passed in 1.14s =====
```