Deployment Diagram - EazZy Project (Versi Final & Valid)

Diagram ini memvisualisasikan arsitektur fisik dan penempatan komponen perangkat lunak (deployment) untuk EazZy Project. Kode PlantUML di bawah ini telah direvisi total menggunakan sintaks yang benar dan standar untuk menghindari error sebelumnya.

Visualisasi Diagram (PlantUML)

Gunakan kode berikut di PlantUML Web Server untuk menghasilkan diagram visual dari arsitektur deployment Anda.

Prompt untuk PlantUML (Versi Final dan Terperbaiki):

```
@startuml
' Diagram Deployment untuk EazZy Project (Versi Final dan Terperbaiki)

!theme plain
title EazZy Project - Deployment Architecture

skinparam rectangle {
  roundCorner 10
}
skinparam node {
  BackgroundColor #F6F6F6
  borderColor #333333
}
skinparam component {
  BackgroundColor #LightSkyBlue
  borderColor #333333
}
skinparam database {
  BackgroundColor #FFFFE0
}

cloud "Internet" as Internet

node "Debian Server (Host)" {
  folder "Host Filesystem" {
    component "[/opt/eazzy-docker]" as ProjectFolder
    component "[/var/lib/docker/volumes]" as DockerVolumes
  }
```

```
    node "Docker Engine" {
      component "[frontend-net]" as FrontendNet
      component "[backend-net]" as BackendNet

      node "Container: reverse-proxy" as NginxContainer {
        component "[Nginx]"
      }

      node "Container: main-app" as MainAppContainer {
        component "[PHP-FPM]"
      }

      database "Container: postgres-db" as DbContainer {
        component "[PostgreSQL]"
      }

      database "Container: redis-cache" as RedisContainer {
        component "[Redis]"
      }

      node "Container: grafana" as GrafanaContainer {
        component "[Grafana]"
      }

      node "Container: prometheus" as PrometheusContainer {
        component "[Prometheus]"
      }
    }
}

' --- Definisi Hubungan ---

' Aliran data dari luar
Internet --> NginxContainer : "HTTPS (Port 80/443)"

' Komunikasi Internal via Jaringan Docker
NginxContainer --> MainAppContainer : "fastcgi_pass (via backend-net)"
MainAppContainer --> DbContainer : "SQL (via backend-net)"
MainAppContainer --> RedisContainer : "Cache (via backend-net)"
PrometheusContainer --> MainAppContainer : "Scrape Metrics (via backend-net)"
GrafanaContainer --> PrometheusContainer : "Query Data (via backend-net)"

' Hubungan Jaringan (asosiasi)
```

```
NginxContainer -- FrontendNet
NginxContainer -- BackendNet
MainAppContainer -- BackendNet
DbContainer -- BackendNet
RedisContainer -- BackendNet
GrafanaContainer -- BackendNet
PrometheusContainer -- BackendNet

' Hubungan Volume (Penyimpanan Persisten)
ProjectFolder ..> NginxContainer : "mounts config"
ProjectFolder ..> MainAppContainer : "mounts code"
DockerVolumes ..> DbContainer : "mounts data volume"
DockerVolumes ..> RedisContainer : "mounts data volume"

@enduml
```

Penjelasan Perbaikan Utama

1. **Struktur yang Disederhanakan:** Saya tidak lagi menumpuk (nesting) artifact di dalam node yang kompleks. Sebaliknya, saya menggunakan elemen standar:
    - node: Untuk lingkungan eksekusi (Server Host, Docker Engine, Kontainer).
    - component: Untuk komponen perangkat lunak atau file.
    - database: Untuk komponen database.
    - folder: Untuk merepresentasikan direktori di filesystem.
2. **Sintaks yang Benar:** Struktur ini sepenuhnya mematuhi standar PlantUML untuk Deployment Diagram, yang seharusnya menghilangkan semua "error type component".
3. **Kejelasan Hubungan:** Hubungan antar komponen (seperti mounts config atau fastcgi_pass) sekarang didefinisikan dengan lebih jelas dan logis.

Sekali lagi, saya mohon maaf atas kesalahan sebelumnya. Kode ini seharusnya menjadi versi final yang solid dan dapat Anda gunakan.