tubes komstat

kelompok 3

2025-11-18

“Pengembangan Fungsi R untuk Analisis Sebaran Spasial-Temporal dan Clustering Hotspot Karhutla di Provinsi Sumatera Selatan (Januari - November 2025)”

```
#library yang digunakan
pkgs <- c("sf", "dplyr", "ggplot2", "lubridate", "leaflet", "cluster",
          "RColorBrewer", "htmlwidgets")

for(pkg in pkgs) {
  if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
    install.packages(pkg, dependencies = TRUE)
    library(pkg, character.only = TRUE)
  }
}
```

```
## Warning: package 'sf' was built under R version 4.4.3
```

```
## Linking to GEOS 3.13.0, GDAL 3.10.1, PROJ 9.5.1; sf_use_s2() is TRUE
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Warning: package 'leaflet' was built under R version 4.4.3
```

```
## Warning: package 'htmlwidgets' was built under R version 4.4.2
```

KATEGORI 1: DATA WRANGLING FUNCTIONS

```
#' Load dan Clean Data Hotspot VIIRS
#' @param csv_path Path ke file CSV FIRMS NASA
#' @param frp_threshold Threshold FRP minimum (MW)
#' @param confidence_min Confidence minimum ("low", "nominal", "high")
#' @return Data frame hotspot yang sudah diproses
load_and_clean_hotspot <- function(csv_path, frp_threshold = 0.5,
                                   confidence_min = "low") {

  if(!file.exists(csv_path)) stop("File CSV tidak ditemukan!")

  hotspot <- read.csv(csv_path, stringsAsFactors = FALSE)

  message(sprintf("Raw data: %d rows", nrow(hotspot)))
  message(sprintf("Confidence values: %s", paste(unique(hotspot$confidence), collapse = ",
")))

  # Map confidence: user input (low/nominal/high) -> FIRMS format (l/n/h)
  conf_levels_full <- c("low", "nominal", "high")
  conf_levels_short <- c("l", "n", "h")

  conf_idx <- match(confidence_min, conf_levels_full)
  if(is.na(conf_idx)) stop("confidence_min harus 'low', 'nominal', atau 'high'")

  valid_conf <- conf_levels_short[conf_idx:3]

  message(sprintf("Filter: confidence >= %s (codes: %s), FRP >= %.2f MW",
                  confidence_min, paste(valid_conf, collapse = ", "), frp_threshold))

  # Filter data
  hotspot <- hotspot %>%
    filter(
      confidence %in% valid_conf,
      !is.na(frp),
      frp >= frp_threshold,
      !is.na(latitude),
      !is.na(longitude)
    )

  message(sprintf("After filtering: %d rows", nrow(hotspot)))

  if(nrow(hotspot) == 0) {
    stop("ERROR: Semua data ter-filter!\n",
         "Coba: frp_threshold = 0.1, confidence_min = 'low'")
  }

  # Process temporal features
  hotspot <- hotspot %>%
    mutate(
      acq_datetime = ymd_hm(paste(acq_date, sprintf("%04d", acq_time))),
      month = month(acq_datetime, label = TRUE, abbr = FALSE),
      month_num = month(acq_datetime),
      year = year(acq_datetime),
      hour = hour(acq_datetime),
      day = day(acq_datetime),
      week = week(acq_datetime),
      month_year = paste(month, year),
      is_night = (daynight == "N")
    ) %>%
    arrange(acq_datetime)

  message(sprintf("✓ Data loaded: %d hotspots (%s to %s)",
                  nrow(hotspot), min(hotspot$acq_date), max(hotspot$acq_date)))

  return(hotspot)
}


#' Convert ke Spatial Object
#' @param hotspot_data Data frame hotspot
#' @param crs Coordinate Reference System (default: 4326)
#' @return SF object
convert_to_spatial <- function(hotspot_data, crs = 4326) {

  if(!inherits(hotspot_data, "data.frame")) {
    stop("hotspot_data harus berupa data frame")
  }

  if(!all(c("longitude", "latitude") %in% names(hotspot_data))) {
    stop("Data harus memiliki kolom 'longitude' dan 'latitude'")
  }

  # Check coordinate validity
  invalid <- hotspot_data %>%
    filter(is.na(longitude) | is.na(latitude) |
           longitude < -180 | longitude > 180 |
           latitude < -90 | latitude > 90)

  if(nrow(invalid) > 0) {
    warning(sprintf("%d rows memiliki koordinat tidak valid", nrow(invalid)))
  }

  hotspot_sf <- st_as_sf(hotspot_data,
                         coords = c("longitude", "latitude"),
                         crs = crs,
                         remove = FALSE)

  message(sprintf("✓ Spatial object created: %d points (CRS: EPSG:%d)",
                  nrow(hotspot_sf), crs))

  return(hotspot_sf)
}


#' Load Shapefile dan Spatial Join (Opsional)
#' @param hotspot_sf Spatial object
#' @param shp_path Path ke shapefile (NULL jika tidak ada)
#' @return List dengan hotspot_sf dan regional summary
join_with_shapefile <- function(hotspot_sf, shp_path = NULL) {
  if(!inherits(hotspot_sf, "sf")) {
    stop("hotspot_sf harus berupa sf object. Gunakan convert_to_spatial() dulu.")
  }

  if(is.null(shp_path) || shp_path == "" || !file.exists(shp_path)) {
    message("⚠ Shapefile tidak tersedia, skip regional analysis")
    return(list(
      hotspot_sf = hotspot_sf,
      regional = NULL,
      region_col = NULL,
      boundary = NULL,
      has_shp = FALSE
    ))
  }

  tryCatch({
    message("Loading shapefile...")
    shp <- st_read(shp_path, quiet = TRUE)

    message(sprintf("  Shapefile: %d features", nrow(shp)))
    message(sprintf("  Columns: %s", paste(names(shp)[names(shp) != "geometry"], collapse =
", ")))

    # Transform CRS if needed
    if(st_crs(shp) != st_crs(hotspot_sf)) {
      message("  Transforming CRS...")
      shp <- st_transform(shp, st_crs(hotspot_sf))
    }

    # Spatial join
    message("  Performing spatial join...")
    hotspot_joined <- st_join(hotspot_sf, shp, join = st_intersects)

    # Auto-detect region column
    poss_cols <- c("NAMOBJ", "NAME", "NAMA", "KABUPATEN", "KECAMATAN",
                   "PROVINSI", "DISTRICT", "CITY", "REGION")
    avail_cols <- names(shp)[names(shp) != "geometry"]
    region_col <- NULL

    for(col in avail_cols) {
      if(toupper(col) %in% poss_cols) {
        region_col <- col
        break
      }
    }

    if(is.null(region_col) && length(avail_cols) > 0) {
      region_col <- avail_cols[1]
      message(sprintf("  Using first column as region: '%s'", region_col))
    }

    # Regional summary
    if(!is.null(region_col)) {
      regional <- hotspot_joined %>%
        st_drop_geometry() %>%
        filter(!is.na(get(region_col))) %>%
        group_by(across(all_of(region_col))) %>%
        summarise(
          total_hotspot = n(),
          avg_brightness = mean(brightness, na.rm = TRUE),
          avg_frp = mean(frp, na.rm = TRUE),
          max_frp = max(frp, na.rm = TRUE),
          .groups = "drop"
        ) %>%
        arrange(desc(total_hotspot))

      n_matched <- sum(!is.na(hotspot_joined[[region_col]]))
      pct_matched <- 100 * n_matched / nrow(hotspot_joined)

      message(sprintf("✓ Regional analysis: %d areas identified", nrow(regional)))
      message(sprintf("  Matched: %d hotspots (%.1f%%)", n_matched, pct_matched))
      message(sprintf("  Region column: '%s'", region_col))

      return(list(
        hotspot_sf = hotspot_joined,
        regional = regional,
        region_col = region_col,
        boundary = shp,
        has_shp = TRUE
      ))
    } else {
      message("⚠ No suitable region column found")
    }

  }, error = function(e) {
    message(sprintf("⚠ Error loading shapefile: %s", e$message))
  })

  return(list(
    hotspot_sf = hotspot_sf,
    regional = NULL,
    region_col = NULL,
    boundary = NULL,
    has_shp = FALSE
  ))
}


#' Create Spatial Grid untuk Density Analysis
#' @param hotspot_sf Spatial object
#' @param grid_size Ukuran grid dalam derajat (default: 0.1)
#' @return Data frame dengan grid summary
create_spatial_grid <- function(hotspot_sf, grid_size = 0.1) {

  coords <- st_coordinates(hotspot_sf)

  grid_data <- hotspot_sf %>%
    st_drop_geometry() %>%
    mutate(
      grid_lon = floor(longitude / grid_size) * grid_size,
      grid_lat = floor(latitude / grid_size) * grid_size,
      grid_id = paste(grid_lon, grid_lat, sep = "_")
    )

  grid_summary <- grid_data %>%
    group_by(grid_id, grid_lon, grid_lat) %>%
    summarise(
      hotspot_count = n(),
      avg_brightness = mean(brightness, na.rm = TRUE),
      avg_frp = mean(frp, na.rm = TRUE),
      max_frp = max(frp, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    arrange(desc(hotspot_count))

  message(sprintf("✓ Grid created: %d cells (size: %.2f°)",
                  nrow(grid_summary), grid_size))

  return(grid_summary)
}


#' Get Data Summary Statistics
#' @param hotspot_data Data frame hotspot
#' @return List dengan statistik ringkasan
get_data_summary <- function(hotspot_data) {

  summary_stats <- list(
    n_total = nrow(hotspot_data),
    date_range = c(min(hotspot_data$acq_date), max(hotspot_data$acq_date)),
    coord_range = list(
      lon = c(min(hotspot_data$longitude), max(hotspot_data$longitude)),
      lat = c(min(hotspot_data$latitude), max(hotspot_data$latitude))
    ),
    frp_stats = list(
      mean = mean(hotspot_data$frp, na.rm = TRUE),
      median = median(hotspot_data$frp, na.rm = TRUE),
      min = min(hotspot_data$frp, na.rm = TRUE),
      max = max(hotspot_data$frp, na.rm = TRUE)
    ),
    brightness_stats = list(
      mean = mean(hotspot_data$brightness, na.rm = TRUE),
      median = median(hotspot_data$brightness, na.rm = TRUE),
      min = min(hotspot_data$brightness, na.rm = TRUE),
      max = max(hotspot_data$brightness, na.rm = TRUE)
    ),
    confidence_dist = table(hotspot_data$confidence),
    daynight_dist = table(hotspot_data$is_night),
    satellites = unique(hotspot_data$satellite)
  )

  return(summary_stats)
}
```

KATEGORI 2: TEMPORAL ANALYSIS FUNCTIONS

```r
#' Analisis Distribusi Bulanan Hotspot
#' @param hotspot_data Data frame hotspot
#' @return Data frame dengan ringkasan bulanan
analyze_monthly_distribution <- function(hotspot_data) {

  monthly <- hotspot_data %>%
    group_by(year, month, month_num) %>%
    summarise(
      total_hotspot = n(),
      avg_brightness = mean(brightness, na.rm = TRUE),
      avg_frp = mean(frp, na.rm = TRUE),
      max_frp = max(frp, na.rm = TRUE),
      high_confidence = sum(confidence == "high"),
      night_detections = sum(is_night),
      .groups = "drop"
    ) %>%
    mutate(
      month_year = paste(month, year),
      pct_night = round(100 * night_detections / total_hotspot, 1)
    ) %>%
    arrange(year, month_num)

  message(sprintf("✓ Monthly analysis: %d months | Peak: %s (%d hotspots)",
                  nrow(monthly),
                  monthly$month_year[which.max(monthly$total_hotspot)],
                  max(monthly$total_hotspot)))

  return(monthly)
}


#' Analisis Pola Hourly
#' @param hotspot_data Data frame hotspot
#' @return Data frame dengan ringkasan per jam
analyze_hourly_pattern <- function(hotspot_data) {

  hourly <- hotspot_data %>%
    group_by(hour) %>%
    summarise(
      total_hotspot = n(),
      avg_brightness = mean(brightness, na.rm = TRUE),
      avg_frp = mean(frp, na.rm = TRUE),
      .groups = "drop"
    )

  message(sprintf("✓ Hourly analysis: Peak at %02d:00 UTC (%d hotspots)",
                  hourly$hour[which.max(hourly$total_hotspot)],
                  max(hourly$total_hotspot)))

  return(hourly)
}
```

KATEGORI 3: CLUSTERING FUNCTIONS

```
#' K-Means Clustering Hotspot
#' @param hotspot_sf Spatial object
#' @param k Jumlah cluster (default: 3)
#' @param nstart Jumlah random starts (default: 25)
#' @return List dengan hasil clustering
perform_kmeans_clustering <- function(hotspot_sf, k = 3, nstart = 25) {

  # Extract coordinates
  coords <- st_coordinates(hotspot_sf)

  # K-Means
  set.seed(123)
  km <- kmeans(coords, centers = k, nstart = nstart, iter.max = 100)

  # Add cluster labels
  hotspot_clustered <- hotspot_sf %>%
    mutate(cluster = as.factor(km$cluster))

  # Cluster statistics
  cluster_stats <- hotspot_clustered %>%
    st_drop_geometry() %>%
    group_by(cluster) %>%
    summarise(
      n_hotspots = n(),
      pct_total = round(100 * n() / nrow(hotspot_clustered), 1),
      center_lon = mean(longitude),
      center_lat = mean(latitude),
      avg_brightness = mean(brightness, na.rm = TRUE),
      avg_frp = mean(frp, na.rm = TRUE),
      max_frp = max(frp, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    arrange(desc(n_hotspots))

  # Quality metrics
  between_ss_ratio <- 100 * km$betweenss / km$totss

  message(sprintf("✓ K-Means clustering: k=%d | Between/Total SS = %.1f%%",
                  k, between_ss_ratio))

  return(list(
    clustered_data = hotspot_clustered,
    model = km,
    stats = cluster_stats,
    coords = coords,
    k = k,
    between_ss_ratio = between_ss_ratio
  ))
}


#' Menentukan K Optimal dengan Elbow Method
#' @param hotspot_sf Spatial object
```

```r
#' @param max_k Maksimum k yang ditest (default: 10)
#' @return List dengan WSS values dan suggested k
find_optimal_k <- function(hotspot_sf, max_k = 10) {

  coords <- st_coordinates(hotspot_sf)
  wss <- numeric(max_k)

  message("Finding optimal k...")
  for(i in 1:max_k) {
    set.seed(123)
    wss[i] <- kmeans(coords, centers = i, nstart = 10)$tot.withinss
  }

  # Simple elbow detection
  if(max_k >= 3) {
    second_diff <- diff(diff(wss))
    optimal_k <- which.max(abs(second_diff)) + 1
  } else {
    optimal_k <- 2
  }

  message(sprintf("✓ Suggested optimal k: %d", optimal_k))

  return(list(
    k_range = 1:max_k,
    wss = wss,
    optimal_k = optimal_k
  ))
}
```

KATEGORI 4: BOOTSTRAP VALIDATION FUNCTIONS

```
#' Bootstrap Validation untuk Clustering Stability
#' @param hotspot_sf Spatial object
#' @param k Jumlah cluster
#' @param n_boot Jumlah iterasi bootstrap (default: 100)
#' @return List dengan metrik validasi
bootstrap_cluster_validation <- function(hotspot_sf, k = 3, n_boot = 100) {

  coords <- st_coordinates(hotspot_sf)
  n_obs <- nrow(coords)

  # Storage
  wss_scores <- numeric(n_boot)
  between_ss <- numeric(n_boot)

  message(sprintf("Bootstrap validation (n=%d)...", n_boot))
  pb <- txtProgressBar(0, n_boot, style = 3)

  set.seed(123)
  for(i in 1:n_boot) {
    # Bootstrap sample
    boot_idx <- sample(n_obs, n_obs, replace = TRUE)
    boot_coords <- coords[boot_idx, ]

    # K-Means on bootstrap sample
    km_boot <- kmeans(boot_coords, centers = k, nstart = 10)

    # Store metrics
    wss_scores[i] <- km_boot$tot.withinss
    between_ss[i] <- km_boot$betweenss

    setTxtProgressBar(pb, i)
  }
  close(pb)

  # Calculate statistics
  mean_wss <- mean(wss_scores)
  sd_wss <- sd(wss_scores)
  cv <- 100 * sd_wss / mean_wss
  ci_95 <- quantile(wss_scores, c(0.025, 0.975))

  # Stability assessment
  stability <- case_when(
    cv < 5 ~ "Sangat Stabil",
    cv < 10 ~ "Stabil",
    cv < 20 ~ "Cukup Stabil",
    TRUE ~ "Kurang Stabil"
  )

  message(sprintf("\n✓ Bootstrap completed | CV = %.2f%% | %s", cv, stability))

  return(list(
    wss_scores = wss_scores,
    between_ss = between_ss,
    mean_wss = mean_wss,
    sd_wss = sd_wss,
    cv = cv,
    ci_95 = ci_95,
    stability = stability,
    n_boot = n_boot
  ))
}
```

KATEGORI 5: VISUALIZATION FUNCTIONS

```
#' Buat Peta Interaktif Leaflet
#' @param clustering_result Hasil dari perform_kmeans_clustering
#' @return Leaflet map object
create_interactive_map <- function(clustering_result) {

  hotspot_sf <- clustering_result$clustered_data

  pal <- colorFactor("Set1", hotspot_sf$cluster)

  map <- leaflet(hotspot_sf) %>%
    addProviderTiles(providers$OpenStreetMap) %>%
    addCircleMarkers(
      radius = 4,
      color = ~pal(cluster),
      fillOpacity = 0.7,
      stroke = TRUE,
      weight = 1,
      popup = ~paste0(
        "<b>Cluster ", cluster, "</b><br>",
        "Date: ", acq_date, "<br>",
        "Time: ", sprintf("%04d", acq_time), " UTC<br>",
        "FRP: ", round(frp, 2), " MW<br>",
        "Brightness: ", round(brightness, 1), " K<br>",
        "Confidence: ", confidence
      )
    ) %>%
    addLegend("bottomright", pal = pal, values = ~cluster,
              title = "Cluster", opacity = 1)

  message("✓ Interactive map created")

  return(map)
}


#' Plot Tren Bulanan
#' @param monthly_data Hasil dari analyze_monthly_distribution
#' @return ggplot object
plot_monthly_trend <- function(monthly_data) {

  monthly_data <- monthly_data %>%
    mutate(month_year = factor(month_year, levels = unique(month_year)))

  p <- ggplot(monthly_data, aes(x = month_year, y = total_hotspot, group = 1)) +
    geom_line(color = "#E74C3C", size = 1.2) +
    geom_point(color = "#C0392B", size = 3) +
    geom_hline(yintercept = mean(monthly_data$total_hotspot),
               linetype = "dashed", color = "gray40", size = 0.8) +
    theme_minimal(base_size = 12) +
    theme(
      axis.text.x = element_text(angle = 45, hjust = 1),
      plot.title = element_text(face = "bold", size = 14),
      panel.grid.minor = element_blank()
```

```r
    ) +
    labs(
      title = "Distribusi Bulanan Hotspot Karhutla",
      subtitle = sprintf("Sumatera Selatan | Rata-rata: %.0f hotspot/bulan",
                        mean(monthly_data$total_hotspot)),
      x = "Bulan",
      y = "Jumlah Hotspot"
    )

  return(p)
}


#' Plot Pola Hourly
#' @param hourly_data Hasil dari analyze_hourly_pattern
#' @return ggplot object
plot_hourly_pattern <- function(hourly_data) {

  p <- ggplot(hourly_data, aes(x = hour, y = total_hotspot)) +
    geom_col(fill = "#3498DB", alpha = 0.8) +
    geom_line(aes(y = total_hotspot), color = "#2C3E50", size = 1) +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", size = 14)) +
    labs(
      title = "Pola Deteksi Hotspot per Jam",
      subtitle = "Distribusi 24 jam (UTC)",
      x = "Jam (UTC)",
      y = "Jumlah Hotspot"
    ) +
    scale_x_continuous(breaks = seq(0, 23, 2))

  return(p)
}


#' Plot Hasil Clustering
#' @param clustering_result Hasil dari perform_kmeans_clustering
#' @return ggplot object
plot_cluster_spatial <- function(clustering_result) {

  coords <- clustering_result$coords
  cluster_df <- data.frame(
    lon = coords[,1],
    lat = coords[,2],
    cluster = clustering_result$clustered_data$cluster
  )

  centers <- clustering_result$stats %>%
    select(cluster, center_lon, center_lat)

  p <- ggplot(cluster_df, aes(x = lon, y = lat, color = cluster)) +
    geom_point(alpha = 0.5, size = 2) +
    geom_point(data = centers, aes(x = center_lon, y = center_lat),
               color = "black", size = 6, shape = 3, stroke = 2) +
    scale_color_brewer(palette = "Set1") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", size = 14)) +
    labs(
      title = "K-Means Clustering Hotspot Karhutla",
      subtitle = sprintf("K = %d clusters | Between/Total SS = %.1f%% | + = centers",
                        clustering_result$k,
                        clustering_result$between_ss_ratio),
      x = "Longitude",
      y = "Latitude",
      color = "Cluster"
    ) +
    coord_fixed(ratio = 1)

  return(p)
}


#' Plot Bootstrap Validation
#' @param bootstrap_result Hasil dari bootstrap_cluster_validation
#' @return ggplot object
plot_bootstrap_stability <- function(bootstrap_result) {

  boot_df <- data.frame(wss = bootstrap_result$wss_scores)

  p <- ggplot(boot_df, aes(x = wss)) +
    geom_histogram(aes(y = after_stat(density)), bins = 30,
                   fill = "#3498DB", alpha = 0.7, color = "white") +
    geom_density(color = "#2C3E50", size = 1.2) +
    geom_vline(xintercept = bootstrap_result$mean_wss,
               color = "#E74C3C", linetype = "dashed", size = 1) +
    geom_vline(xintercept = bootstrap_result$ci_95,
               color = "#27AE60", linetype = "dotted", size = 1) +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", size = 14)) +
    labs(
      title = "Bootstrap Validation: Stabilitas Clustering",
      subtitle = sprintf(
        "n = %d iterations | CV = %.2f%% | Stabilitas: %s",
        bootstrap_result$n_boot,
        bootstrap_result$cv,
        bootstrap_result$stability
      ),
      x = "Within-Cluster Sum of Squares",
      y = "Density",
      caption = "Garis merah = mean | Garis hijau = 95% CI"
    )

  return(p)
}


#' Plot Elbow Method
#' @param elbow_result Hasil dari find_optimal_k
#' @return ggplot object
plot_elbow_method <- function(elbow_result) {

  elbow_df <- data.frame(
    k = elbow_result$k_range,
    wss = elbow_result$wss
  )

  p <- ggplot(elbow_df, aes(x = k, y = wss)) +
    geom_line(color = "#2C3E50", size = 1.2) +
    geom_point(color = "#E74C3C", size = 3) +
    geom_vline(xintercept = elbow_result$optimal_k,
               linetype = "dashed", color = "#27AE60", size = 1) +
    annotate("text",
             x = elbow_result$optimal_k,
             y = max(elbow_df$wss) * 0.9,
             label = sprintf("Optimal k = %d", elbow_result$optimal_k),
             hjust = -0.1, color = "#27AE60", fontface = "bold") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", size = 14)) +
    labs(
      title = "Elbow Method: Penentuan K Optimal",
      subtitle = "Within-Cluster Sum of Squares vs K",
      x = "Number of Clusters (k)",
      y = "Within-Cluster SS"
    ) +
    scale_x_continuous(breaks = elbow_result$k_range)

  return(p)
}


#' Plot Regional Barplot (jika ada shapefile)
#' @param regional_data Regional summary dari join_with_shapefile
#' @param region_col Nama kolom region
#' @param top_n Jumlah top regions yang ditampilkan
#' @return ggplot object
plot_regional_distribution <- function(regional_data, region_col, top_n = 15) {

  if(is.null(regional_data)) return(NULL)

  top_regions <- head(regional_data, top_n)

  p <- ggplot(top_regions,
              aes(x = reorder(get(region_col), total_hotspot),
                  y = total_hotspot)) +
    geom_col(fill = "#E67E22", alpha = 0.8) +
    coord_flip() +
    theme_minimal(base_size = 11) +
    theme(plot.title = element_text(face = "bold", size = 14)) +
    labs(
      title = sprintf("Top %d Wilayah dengan Hotspot Terbanyak", top_n),
      subtitle = "Sumatera Selatan",
      x = "Wilayah",
      y = "Jumlah Hotspot"
    )

  return(p)
}
```

```r
#' Fungsi Utama: Analisis Karhutla Lengkap
#' @param csv_path Path ke file CSV hotspot VIIRS
#' @param shp_path Path ke shapefile (opsional, NULL jika tidak ada)
#' @param k Jumlah cluster untuk K-Means (default: 3)
#' @param n_boot Jumlah iterasi bootstrap (default: 100)
#' @param frp_threshold Threshold FRP minimum MW (default: 0.5)
#' @param confidence_min Confidence minimum ("low", "nominal", "high")
#' @param find_k_optimal Apakah mencari k optimal dulu? (default: FALSE)
#' @param max_k Maksimum k untuk elbow method (default: 10)
#' @param output_dir Direktori output (default: "hasil_karhutla")
#' @return List lengkap berisi semua hasil analisis
analyze_karhutla <- function(csv_path,
                             shp_path = NULL,
                             k = 3,
                             n_boot = 100,
                             frp_threshold = 0.5,
                             confidence_min = "low",
                             find_k_optimal = FALSE,
                             max_k = 10,
                             output_dir = "hasil_karhutla") {

  cat("\n")
  cat("================================================================================\n")
  cat("  ANALISIS HOTSPOT KARHUTLA SUMATERA SELATAN\n")
  cat("  Pengembangan Fungsi R untuk Data VIIRS FIRMS\n")
  cat("================================================================================\n\n")

  start_time <- Sys.time()

  # ========== STEP 1: DATA WRANGLING ==========
  cat("[STEP 1/7] DATA WRANGLING\n")
  cat("--------------------------------------------------------------------------------\n")

  hotspot <- load_and_clean_hotspot(csv_path, frp_threshold, confidence_min)
  hotspot_sf <- convert_to_spatial(hotspot)
  spatial_result <- join_with_shapefile(hotspot_sf, shp_path)

  cat("\n")

  # ========== STEP 2: TEMPORAL ANALYSIS ==========
  cat("[STEP 2/7] TEMPORAL ANALYSIS\n")
  cat("--------------------------------------------------------------------------------\n")

  monthly <- analyze_monthly_distribution(hotspot)
  hourly <- analyze_hourly_pattern(hotspot)

  cat("\n")

  # ========== STEP 3: FIND OPTIMAL K (OPSIONAL) ==========
  elbow_result <- NULL
  if(find_k_optimal) {
    cat("[STEP 3/7] FINDING OPTIMAL K\n")
    cat("--------------------------------------------------------------------------------\n")

    elbow_result <- find_optimal_k(spatial_result$hotspot_sf, max_k)
    k_suggested <- elbow_result$optimal_k

    cat(sprintf("\n⚠ Suggested k = %d (you can change this)\n", k_suggested))
    cat(sprintf("⚠ Using k = %d for clustering\n\n", k))
  } else {
    cat("[STEP 3/7] SKIPPING OPTIMAL K SEARCH\n")
    cat("--------------------------------------------------------------------------------\n")
    cat(sprintf("Using user-specified k = %d\n\n", k))
  }

  # ========== STEP 4: K-MEANS CLUSTERING ==========
  cat("[STEP 4/7] K-MEANS CLUSTERING\n")
  cat("--------------------------------------------------------------------------------\n")

  clustering <- perform_kmeans_clustering(spatial_result$hotspot_sf, k)

  cat("\nCluster Distribution:\n")
  print(clustering$stats)
  cat("\n")

  # ========== STEP 5: BOOTSTRAP VALIDATION ==========
  cat("[STEP 5/7] BOOTSTRAP VALIDATION\n")
  cat("--------------------------------------------------------------------------------\n")

  validation <- bootstrap_cluster_validation(spatial_result$hotspot_sf, k, n_boot)

  cat("\n")

  # ========== STEP 6: VISUALIZATION ==========
  cat("[STEP 6/7] CREATING VISUALIZATIONS\n")
  cat("--------------------------------------------------------------------------------\n")

  message("Creating interactive map...")
  map <- create_interactive_map(clustering)

  message("Creating monthly trend plot...")
  p_monthly <- plot_monthly_trend(monthly)

  message("Creating hourly pattern plot...")
  p_hourly <- plot_hourly_pattern(hourly)

  message("Creating cluster plot...")
  p_cluster <- plot_cluster_spatial(clustering)

  message("Creating bootstrap plot...")
  p_bootstrap <- plot_bootstrap_stability(validation)

  p_elbow <- NULL
  if(!is.null(elbow_result)) {
    message("Creating elbow plot...")
    p_elbow <- plot_elbow_method(elbow_result)
  }

  p_regional <- NULL
  if(spatial_result$has_shp) {
    message("Creating regional plot...")
    p_regional <- plot_regional_distribution(
      spatial_result$regional,
      spatial_result$region_col
    )
  }

  cat("\n")

  # ========== STEP 7: SAVE OUTPUTS ==========
  cat("[STEP 7/8] SAVING OUTPUTS\n")
  cat("--------------------------------------------------------------------------------\n")

  # Create output directory
  output_dir <- path.expand(output_dir)
  dir.create(output_dir, showWarnings = FALSE, recursive = TRUE)

  # Save CSV files
  message("Saving CSV files...")
  write.csv(monthly, file.path(output_dir, "monthly_distribution.csv"), row.names = FALSE)
  write.csv(hourly, file.path(output_dir, "hourly_pattern.csv"), row.names = FALSE)
  write.csv(clustering$stats, file.path(output_dir, "cluster_statistics.csv"), row.names = FALSE)

  if(spatial_result$has_shp) {
    write.csv(spatial_result$regional, file.path(output_dir, "regional_summary.csv"), row.names = FALSE)
  }

  # Save validation results
  validation_summary <- data.frame(
    metric = c("Mean WSS", "SD WSS", "CV (%)", "CI 95% Lower", "CI 95% Upper", "Stability"),
    value = as.character(c(
      round(validation$mean_wss, 2),
      round(validation$sd_wss, 2),
      round(validation$cv, 2),
      round(validation$ci_95[1], 2),
      round(validation$ci_95[2], 2),
      validation$stability
    ))
  )
  write.csv(validation_summary, file.path(output_dir, "bootstrap_validation.csv"), row.names = FALSE)

  # Save plots (300 DPI)
  message("Saving plots (300 DPI)...")
  ggsave(file.path(output_dir, "01_monthly_trend.png"), p_monthly, width = 12, height = 6, dpi = 300)
  ggsave(file.path(output_dir, "02_hourly_pattern.png"), p_hourly, width = 10, height = 6, dpi = 300)
  ggsave(file.path(output_dir, "03_cluster_spatial.png"), p_cluster, width = 10, height = 8, dpi = 300)
  ggsave(file.path(output_dir, "04_bootstrap_validation.png"), p_bootstrap, width = 10, height = 6, dpi = 300)

  if(!is.null(p_elbow)) {
    ggsave(file.path(output_dir, "05_elbow_method.png"), p_elbow, width = 10, height = 6, dpi
= 300)
  }

  if(!is.null(p_regional)) {
    ggsave(file.path(output_dir, "06_regional_distribution.png"), p_regional, width = 10, hei
ght = 8, dpi = 300)
  }

  # Save interactive map
  message("Saving interactive map...")
  saveWidget(map, file.path(output_dir, "interactive_map.html"), selfcontained = TRUE)

  # Save R object
  message("Saving R data object...")
  result_obj <- list(
    data = hotspot,
    spatial_data = spatial_result$hotspot_sf,
    temporal = list(monthly = monthly, hourly = hourly),
    clustering = clustering,
    validation = validation,
    regional = spatial_result$regional,
    elbow = elbow_result
  )
  saveRDS(result_obj, file.path(output_dir, "karhutla_analysis.rds"))

  message(sprintf("\n✓ All outputs saved to: %s\n", output_dir))

  cat("\n")

  # ========== STEP 8: DISPLAY IN RSTUDIO ==========
  cat("[STEP 8/8] DISPLAYING RESULTS IN RSTUDIO\n")
  cat("----------------------------------------------------------------------------------\n")

  message("Displaying visualizations in RStudio...")

  # Display plots in Plots panel
  print(p_monthly)
  Sys.sleep(0.5)

  print(p_hourly)
  Sys.sleep(0.5)

  print(p_cluster)
  Sys.sleep(0.5)

  print(p_bootstrap)
  Sys.sleep(0.5)

  if(!is.null(p_elbow)) {
    print(p_elbow)
    Sys.sleep(0.5)
  }

  if(!is.null(p_regional)) {
    print(p_regional)
    Sys.sleep(0.5)
  }

  # Display interactive map in Viewer panel
  message("\nDisplaying interactive map in Viewer...")
  print(map)

  # Display summary tables in Console
  cat("\n")
  cat("═══════════════════════════════════════════════════════════════════════════════\n")
  cat("  CLUSTER STATISTICS\n")
  cat("═══════════════════════════════════════════════════════════════════════════════\n\n")
  print(clustering$stats, row.names = FALSE)

  cat("\n")
  cat("═══════════════════════════════════════════════════════════════════════════════\n")
  cat("  BOOTSTRAP VALIDATION\n")
  cat("═══════════════════════════════════════════════════════════════════════════════\n\n")
  print(validation_summary, row.names = FALSE)

  if(spatial_result$has_shp) {
    cat("\n")
    cat("═══════════════════════════════════════════════════════════════════════════════\n")
    cat("  TOP 10 REGIONAL DISTRIBUTION\n")
    cat("═══════════════════════════════════════════════════════════════════════════════\n
\n")
    print(head(spatial_result$regional, 10), row.names = FALSE)
  }

  cat("\n")
  cat("═══════════════════════════════════════════════════════════════════════════════\n")
  cat("  MONTHLY TREND (Top 5 Months)\n")
  cat("═══════════════════════════════════════════════════════════════════════════════\n\n")
  top_months <- monthly %>%
    arrange(desc(total_hotspot)) %>%
    head(5) %>%
    select(month_year, total_hotspot, avg_frp, high_confidence)
  print(top_months, row.names = FALSE)

  message("\n✓ All results displayed in RStudio!")
  message("  → Plots: Check 'Plots' panel (use arrows to navigate)")
  message("  → Map: Check 'Viewer' panel")
  message("  → Tables: See console output above")
  message(sprintf("  → Files: %s\n", output_dir))

  cat("\n")

  # ========== SUMMARY ==========
  end_time <- Sys.time()
  elapsed <- as.numeric(difftime(end_time, start_time, units = "mins"))

  cat("================================================================================\n")
  cat("  ANALISIS SELESAI!\n")
  cat("================================================================================\n\n")

  cat("RINGKASAN HASIL:\n")
  cat("--------------------------------------------------------------------------------\n")
  cat(sprintf("Dataset:\n"))
  cat(sprintf("  • Total Hotspot: %d\n", nrow(hotspot)))
  cat(sprintf("  • Periode: %s s/d %s\n", min(hotspot$acq_date), max(hotspot$acq_date)))
  cat(sprintf("  • Coordinate Range: Lon [%.3f, %.3f], Lat [%.3f, %.3f]\n",
              min(hotspot$longitude), max(hotspot$longitude),
              min(hotspot$latitude), max(hotspot$latitude)))

  cat(sprintf("\nTemporal Analysis:\n"))
  cat(sprintf("  • Total Bulan: %d\n", nrow(monthly)))
  cat(sprintf("  • Bulan Puncak: %s (%d hotspot)\n",
              monthly$month_year[which.max(monthly$total_hotspot)],
              max(monthly$total_hotspot)))
  cat(sprintf("  • Jam Puncak: %02d:00 UTC (%d hotspot)\n",
              hourly$hour[which.max(hourly$total_hotspot)],
              max(hourly$total_hotspot)))

  cat(sprintf("\nClustering Analysis:\n"))
  cat(sprintf("  • K (jumlah cluster): %d\n", k))
  cat(sprintf("  • Between/Total SS Ratio: %.2f%%\n", clustering$between_ss_ratio))
  cat(sprintf("  • Cluster Terbesar: Cluster %s (%d hotspot, %.1f%%)\n",
              clustering$stats$cluster[1],
              clustering$stats$n_hotspots[1],
              clustering$stats$pct_total[1]))

  cat(sprintf("\nBootstrap Validation:\n"))
  cat(sprintf("  • Iterasi: %d\n", n_boot))
  cat(sprintf("  • Coefficient of Variation: %.2f%%\n", validation$cv))
  cat(sprintf("  • 95%% Confidence Interval: [%.2f, %.2f]\n",
              validation$ci_95[1], validation$ci_95[2]))
  cat(sprintf("  • Level Stabilitas: %s\n", validation$stability))

  if(spatial_result$has_shp) {
    cat(sprintf("\nRegional Analysis:\n"))
    cat(sprintf("  • Jumlah Wilayah: %d\n", nrow(spatial_result$regional)))
    cat(sprintf("  • Wilayah Tertinggi: %s (%d hotspot)\n",
                spatial_result$regional[[spatial_result$region_col]][1],
                spatial_result$regional$total_hotspot[1]))
  }

  cat(sprintf("\nOutput Files:\n"))
  n_csv <- 3 + ifelse(spatial_result$has_shp, 2, 1)
  n_png <- 4 + ifelse(!is.null(p_elbow), 1, 0) + ifelse(!is.null(p_regional), 1, 0)
  cat(sprintf("  • CSV: %d file(s)\n", n_csv))
  cat(sprintf("  • PNG Plots: %d file(s) (300 DPI)\n", n_png))
  cat(sprintf("  • Interactive Map: 1 HTML file\n"))
  cat(sprintf("  • R Objects: 1 RDS file\n"))

  cat(sprintf("\nWaktu Eksekusi: %.1f menit\n", elapsed))
  cat(sprintf("Lokasi Output: %s\n", output_dir))

  cat("\n================================================================================\n
\n")

  # Open output folder
  if(.Platform$OS.type == "windows") {
    shell.exec(output_dir)
  } else if(Sys.info()["sysname"] == "Darwin") {
    system(sprintf("open '%s'", output_dir))
  }

  # Return results
  invisible(result_obj)
}
```

```
# Source script ini
#source("kodetubes-k3.Rmd")

# Jalankan analisis
hasil <- analyze_karhutla(
  csv_path = "D:/tubes-komstat/DL_FIRE_J1V-C2_687296/fire_archive_J1V-C2_687296.csv",
  shp_path = "D:/tubes-komstat/DL_FIRE_J1V-C2_687297/fire_archive_J1V-C2_687297.shp",
  k = 3,
  n_boot = 100,
  find_k_optimal = FALSE,
  output_dir = "D:/tubes-komstat/output"
)
```

```
##
## ================================================================================
##   ANALISIS HOTSPOT KARHUTLA SUMATERA SELATAN
##   Pengembangan Fungsi R untuk Data VIIRS FIRMS
## ================================================================================
##
## [STEP 1/7] DATA WRANGLING
## --------------------------------------------------------------------------------
```

```
## Raw data: 3072 rows
```

```
## Confidence values: n, l, h
```

```
## Filter: confidence >= low (codes: l, n, h), FRP >= 0.50 MW
```

```
## After filtering: 3007 rows
```

```
## ✓ Data loaded: 3007 hotspots (2025-01-02 to 2025-07-31)
```

```
## ✓ Spatial object created: 3007 points (CRS: EPSG:4326)
```

```
## Loading shapefile...
```

```
## Re-reading with feature count reset from 3072 to 3053
```

```
##   Shapefile: 3053 features
##   Columns: LATITUDE, LONGITUDE, BRIGHTNESS, SCAN, TRACK, ACQ_DATE, ACQ_TIME, SATELLITE, IN
STRUMENT, CONFIDENCE, VERSION, BRIGHT_T31, FRP, DAYNIGHT
```

```
##   Performing spatial join...
```

```
##   Using first column as region: 'LATITUDE'
```

```
## ✓ Regional analysis: 2965 areas identified
```

```
##   Matched: 2988 hotspots (99.4%)
```

```
##   Region column: 'LATITUDE'
```

```
## 
## [STEP 2/7] TEMPORAL ANALYSIS
## --------------------------------------------------------------------------------
```

```
## ✓ Monthly analysis: 7 months | Peak: July 2025 (1502 hotspots)
```

```
## ✓ Hourly analysis: Peak at 06:00 UTC (1701 hotspots)
```

```
## 
## [STEP 3/7] SKIPPING OPTIMAL K SEARCH
## --------------------------------------------------------------------------------
## Using user-specified k = 3
## 
## [STEP 4/7] K-MEANS CLUSTERING
## --------------------------------------------------------------------------------
```

```
## ✓ K-Means clustering: k=3 | Between/Total SS = 73.7%
```

```
## 
## Cluster Distribution:
## # A tibble: 3 × 8
##   cluster n_hotspots pct_total center_lon center_lat avg_brightness avg_frp
##   <fct>        <int>     <dbl>      <dbl>      <dbl>          <dbl>   <dbl>
## 1 1             1260      41.9       104.      -3.61           331.    6.38
## 2 3             1006      33.5       103.      -2.17           328.    6.20
## 3 2              741      24.6       106.      -2.36           337.    8.52
## # ℹ 1 more variable: max_frp <dbl>
## 
## [STEP 5/7] BOOTSTRAP VALIDATION
## --------------------------------------------------------------------------------
```

```
## Bootstrap validation (n=100)...
##   |                                                                               |
|   0%  |                                                                               |=
|   1%  |                                                                               |=
|   2%  |                                                                               |==
|   3%  |                                                                               |===
|   4%  |                                                                               |====
|   5%  |                                                                               |====
|   6%  |                                                                               |=====
|   7%  |                                                                               |=====
=                                                                |   8%  |
|======                                                                |   9%  |
|=======                                                               |  10%  |
|========                                                              |  11%  |
|========                                                              |  12%  |
|=========                                                             |  13%  |
|==========                                                            |  14%  |
|==========                                                            |  15%  |
|===========                                                           |  16%  |
|============                                                          |  17%  |
|=============                                                         |  18%  |
|=============                                                         |  19%  |
|==============                                                        |  20%  |
|===============                                                       |  21%  |
|===============                                                       |  22%  |
|================                                                      |  23%  |
|=================                                                     |  24%  |
|==================                                                    |  25%  |
|==================                                                    |  26%  |
|===================                                                   |  27%  |
|====================                                                  |  28%  |
|====================                                                  |  29%  |
|=====================                                                 |  30%  |
|======================                                                |  31%  |
|======================                                                |  32%  |
|=======================                                               |  33%  |
|========================                                              |  34%  |
|========================                                              |  35%  |
|=========================                                             |  36%  |
|==========================                                            |  37%  |
|===========================                                           |  38%  |
|===========================                                           |  39%  |
|============================                                          |  40%  |
|=============================                                         |  41%  |
|=============================                                         |  42%  |
|==============================                                        |  43%  |
|===============================                                       |  44%  |
|================================                                      |  45%  |
|================================                                      |  46%  |
|=================================                                     |  47%  |
|==================================                                    |  48%  |
|==================================                                    |  49%  |
|===================================                                   |  50%  |
|====================================                                  |  51%  |
|====================================                                  |  52%  |
|=====================================                                 |  53%  |
|======================================                                |  54%  |
|======================================                                |  55%  |
|=======================================                               |  56%  |
|========================================                              |  57%  |
|=========================================                             |  58%  |
|=========================================                             |  59%  |
|==========================================                            |  60%  |
|===========================================                           |  61%  |
|===========================================                           |  62%  |
|============================================                          |  63%  |
|=============================================                         |  64%  |
|==============================================                        |  65%  |
|==============================================                        |  66%  |
|===============================================                       |  67%  |
|================================================                      |  68%  |
|================================================                      |  69%  |
|=================================================                     |  70%  |
|==================================================                    |  71%  |
|==================================================                    |  72%  |
|===================================================                   |  73%  |
|====================================================                  |  74%  |
|====================================================                  |  75%  |
|=====================================================                 |  76%  |
|======================================================                |  77%  |
|=======================================================               |  78%  |
|=======================================================               |  79%  |
|========================================================              |  80%  |
|=========================================================             |  81%  |
|=========================================================             |  82%  |
|==========================================================            |  83%  |
|===========================================================           |  84%  |
|============================================================          |  85%  |
|============================================================          |  86%  |
|=============================================================         |  87%  |
|==============================================================        |  88%  |
|==============================================================        |  89%  |
|===============================================================       |  90%  |
|================================================================      |  91%  |
|================================================================      |  92%  |
|=================================================================     |  93%  |
|==================================================================    |  94%  |
|==================================================================    |  95%  |
|===================================================================   |  96%  |
|====================================================================  |  97%  |
|===================================================================== |  98%  |
|===================================================================== |  99%  |
|======================================================================| 100%
```

```
##
## ✓ Bootstrap completed | CV = 1.79% | Sangat Stabil
```

```
##
## [STEP 6/7] CREATING VISUALIZATIONS
## ---------------------------------------------------------------------------------
## Creating interactive map...
```

```
## ✓ Interactive map created
```

```
## Creating monthly trend plot...
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Creating hourly pattern plot...
```

```
## Creating cluster plot...
```

```
## Creating bootstrap plot...
```

```
## Creating regional plot...
```

```
##
## [STEP 7/8] SAVING OUTPUTS
## ----------------------------------------------------------------------------
```

```
## Saving CSV files...
```

```
## Saving plots (300 DPI)...
```

```
## Saving interactive map...
```

```
## Saving R data object...
```

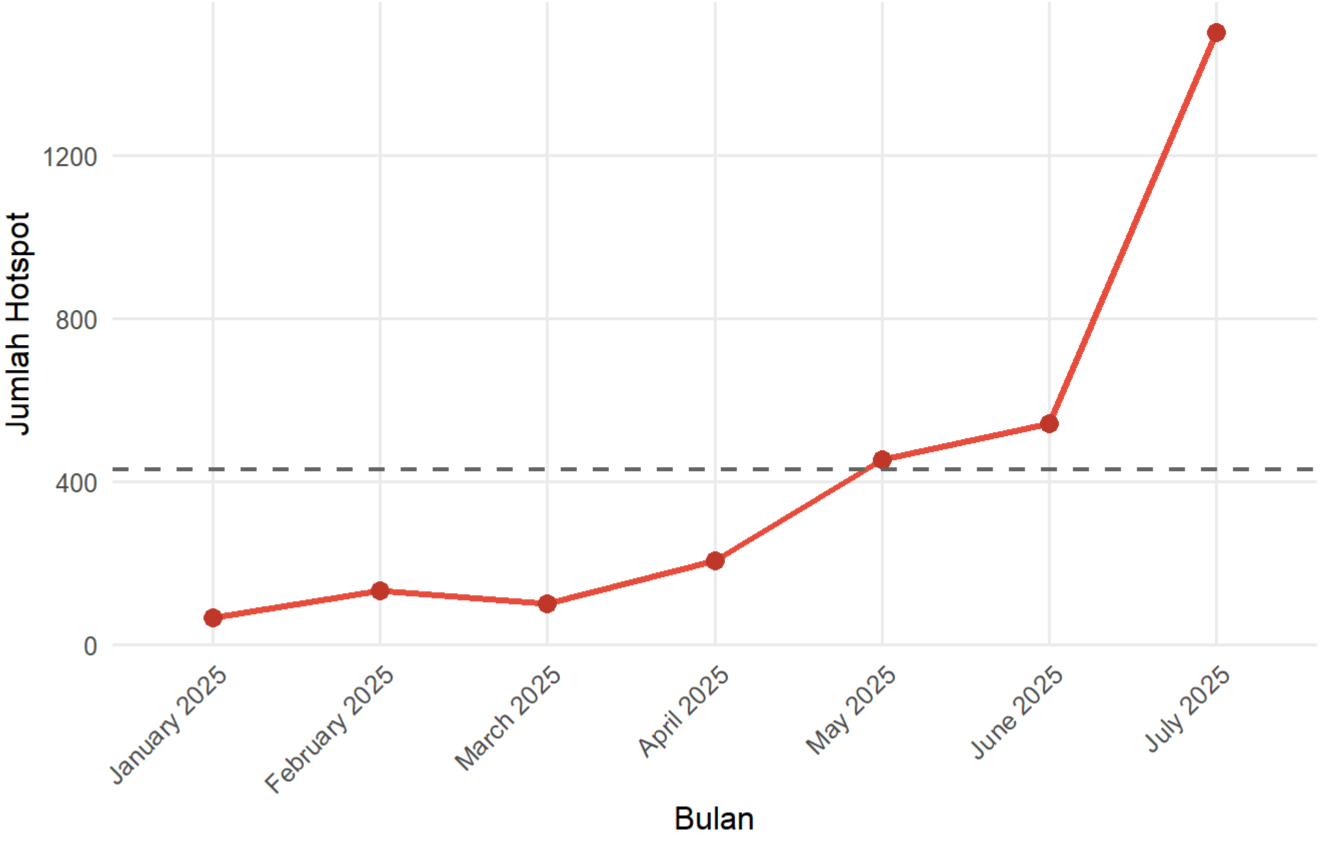
```
##
## ✓ All outputs saved to: D:/tubes-komstat/output
```

```
##
## [STEP 8/8] DISPLAYING RESULTS IN RSTUDIO
## ----------------------------------------------------------------------------
```

```
## Displaying visualizations in RStudio...
```

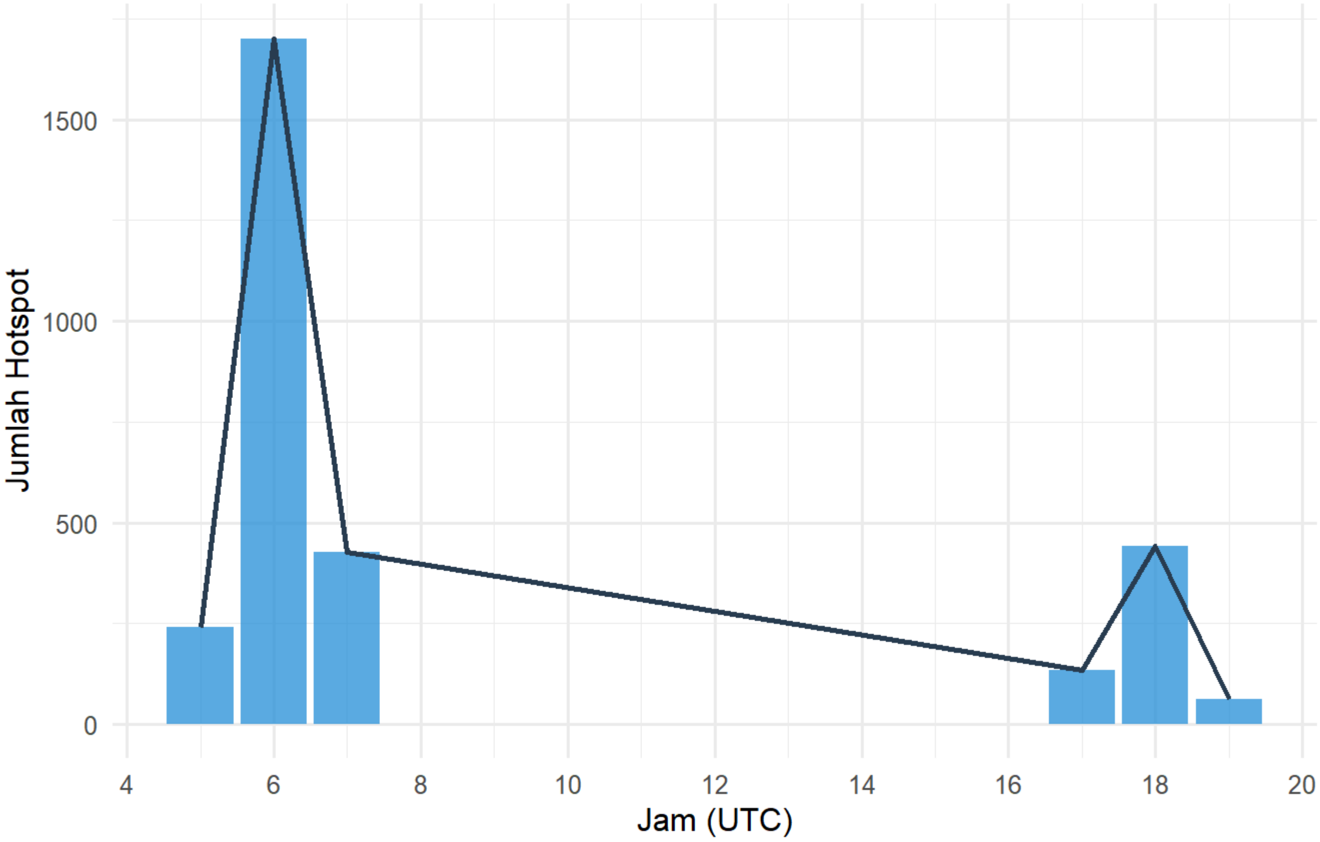
Distribusi Bulanan Hotspot Karhutla

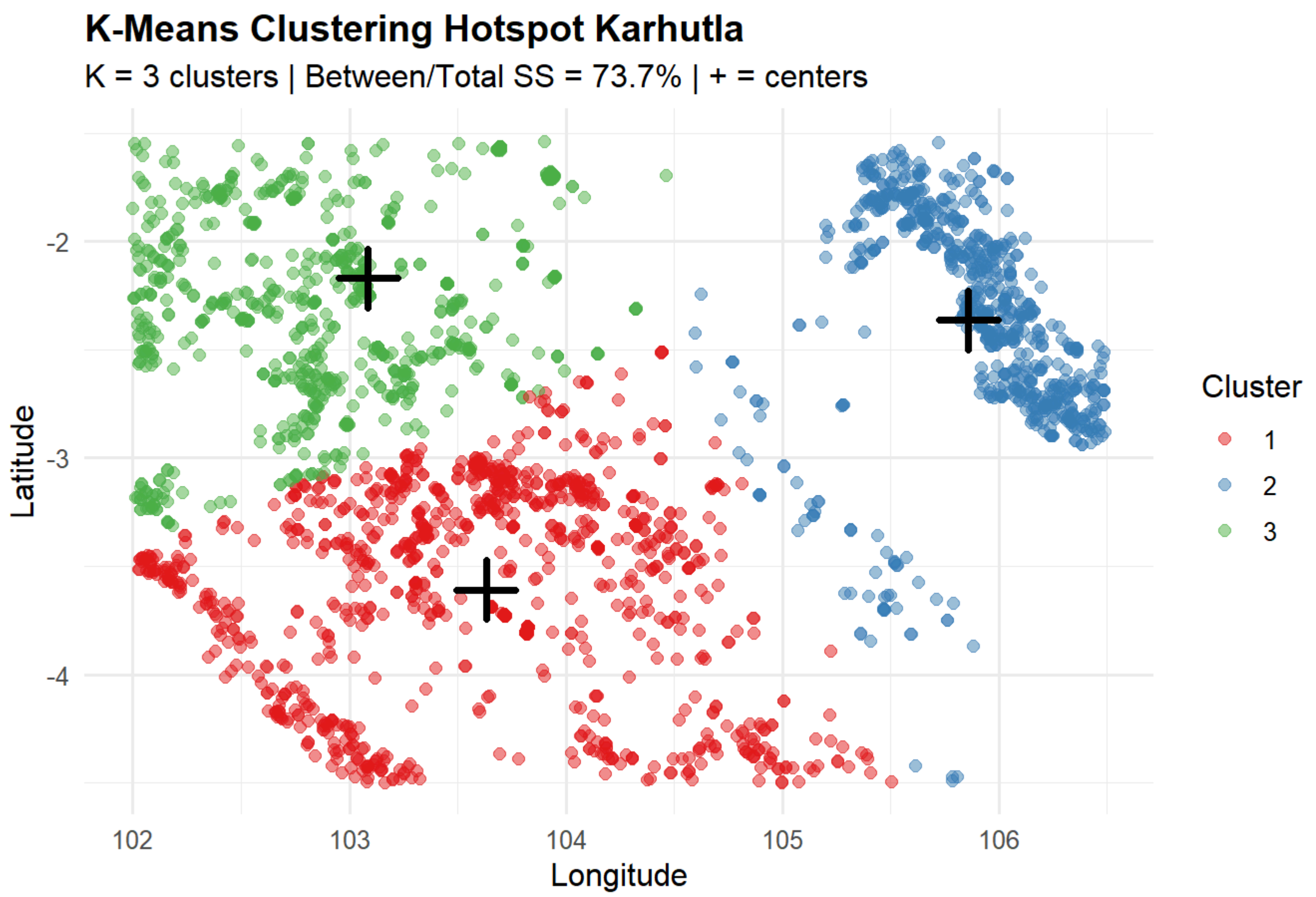
Sumatera Selatan | Rata-rata: 430 hotspot/bulan



Pola Deteksi Hotspot per Jam

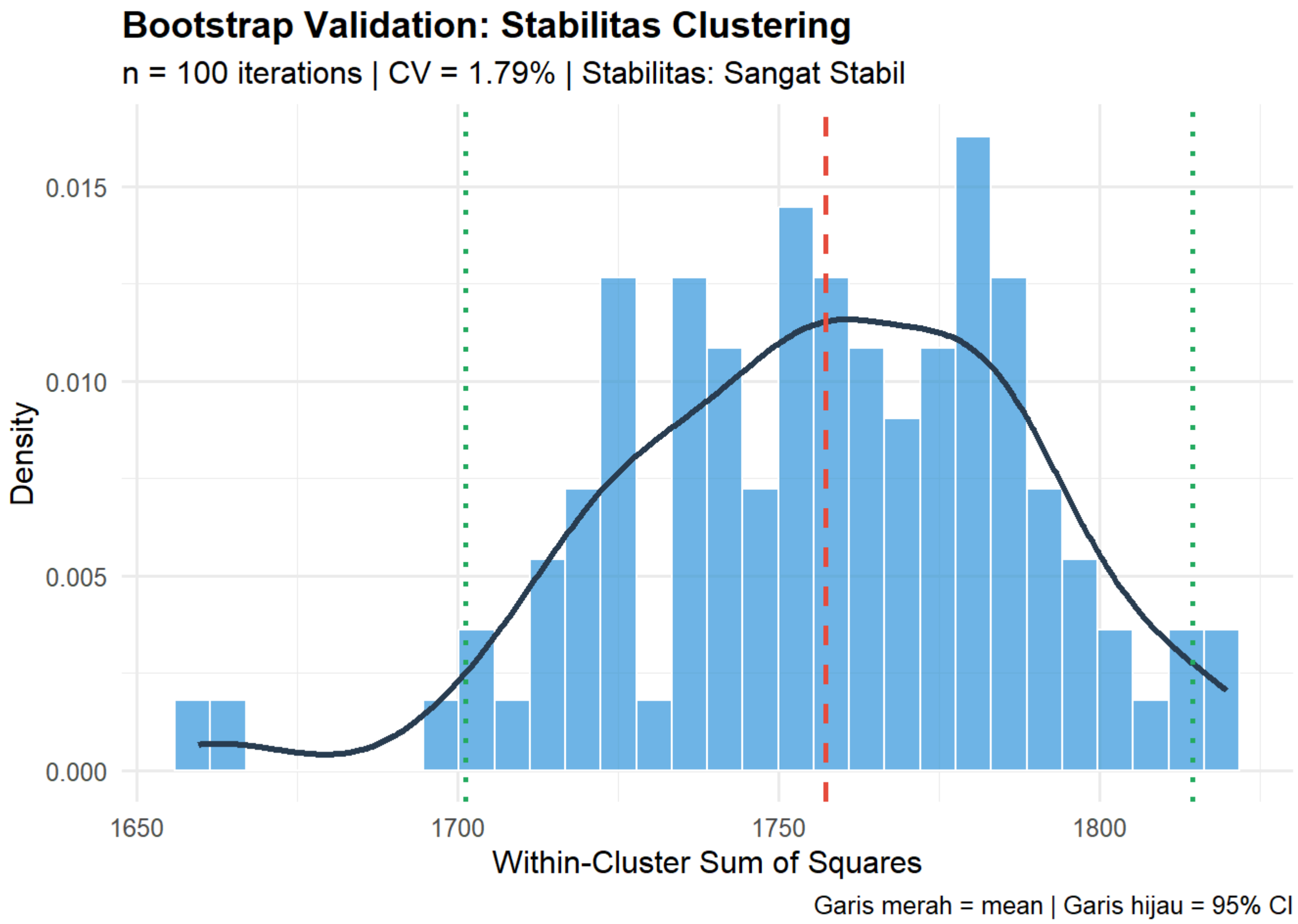
Distribusi 24 jam (UTC)





K-Means Clustering Hotspot Karhutla

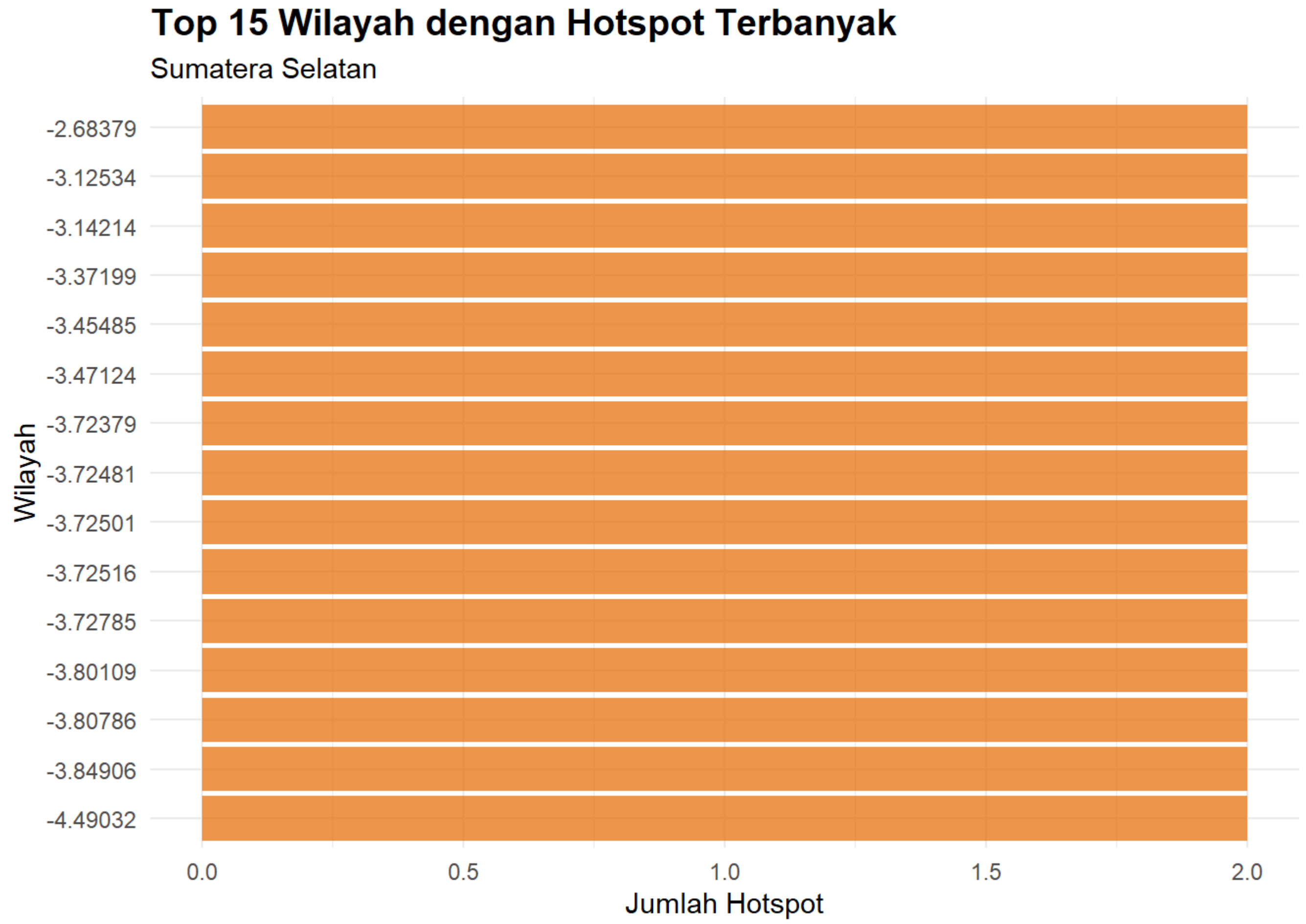
K = 3 clusters | Between/Total SS = 73.7% | + = centers



Bootstrap Validation: Stabilitas Clustering

n = 100 iterations | CV = 1.79% | Stabilitas: Sangat Stabil

Garis merah = mean | Garis hijau = 95% CI



```
##
## Displaying interactive map in Viewer...
```

```
##
## ════════════════════════════════════════════════════════════════════════════════
##   CLUSTER STATISTICS
## ════════════════════════════════════════════════════════════════════════════════
##
## # A tibble: 3 × 8
##   cluster n_hotspots pct_total center_lon center_lat avg_brightness avg_frp
##   <fct>        <int>     <dbl>      <dbl>      <dbl>          <dbl>   <dbl>
## 1 1             1260      41.9       104.      -3.61           331.    6.38
## 2 3             1006      33.5       103.      -2.17           328.    6.20
## 3 2              741      24.6       106.      -2.36           337.    8.52
## # ℹ 1 more variable: max_frp <dbl>
##
## ════════════════════════════════════════════════════════════════════════════════
##   BOOTSTRAP VALIDATION
## ════════════════════════════════════════════════════════════════════════════════
##
##        metric         value
##      Mean WSS       1757.33
##        SD WSS         31.38
##        CV (%)          1.79
##  CI 95% Lower       1701.24
##  CI 95% Upper       1814.43
##     Stability Sangat Stabil
##
## ════════════════════════════════════════════════════════════════════════════════
##   TOP 10 REGIONAL DISTRIBUTION
## ════════════════════════════════════════════════════════════════════════════════
##
## # A tibble: 10 × 5
##    LATITUDE total_hotspot avg_brightness avg_frp max_frp
##       <dbl>         <int>          <dbl>   <dbl>   <dbl>
##  1    -4.49             2           334.    4.03    4.47
##  2    -3.85             2           320.    5.46    6.59
##  3    -3.81             2           330.    3.26    4.37
##  4    -3.80             2           324.    2.72    3.44
##  5    -3.73             2           351.   48.6    73.7
##  6    -3.73             2           329.    2.83    3.94
##  7    -3.73             2           339.    6.98    9.16
##  8    -3.72             2           339.    4.40    4.6
##  9    -3.72             2           335.    4.34    4.52
## 10    -3.47             2           324.    2.57    3.61
##
## ════════════════════════════════════════════════════════════════════════════════
##   MONTHLY TREND (Top 5 Months)
## ════════════════════════════════════════════════════════════════════════════════
##
## # A tibble: 5 × 4
##   month_year    total_hotspot avg_frp high_confidence
##   <chr>                 <int>   <dbl>           <int>
## 1 July 2025              1502    7.80               0
## 2 June 2025               542    6.12               0
## 3 May 2025                454    6.31               0
## 4 April 2025              206    4.90               0
## 5 February 2025           134    5.00               0
```

```
##
## ✓ All results displayed in RStudio!
```

```
##   → Plots: Check 'Plots' panel (use arrows to navigate)
```

```
##   → Map: Check 'Viewer' panel
```

```
##   → Tables: See console output above
```

```
##   → Files: D:/tubes-komstat/output
```

```
## 
## ================================================================================
##   ANALISIS SELESAI!
## ================================================================================
## 
## RINGKASAN HASIL:
## --------------------------------------------------------------------------------
## Dataset:
##   • Total Hotspot: 3007
##   • Periode: 2025-01-02 s/d 2025-07-31
##   • Coordinate Range: Lon [102.000, 106.490], Lat [-4.500, -1.535]
## 
## Temporal Analysis:
##   • Total Bulan: 7
##   • Bulan Puncak: July 2025 (1502 hotspot)
##   • Jam Puncak: 06:00 UTC (1701 hotspot)
## 
## Clustering Analysis:
##   • K (jumlah cluster): 3
##   • Between/Total SS Ratio: 73.71%
##   • Cluster Terbesar: Cluster 1 (1260 hotspot, 41.9%)
## 
## Bootstrap Validation:
##   • Iterasi: 100
##   • Coefficient of Variation: 1.79%
##   • 95% Confidence Interval: [1701.24, 1814.43]
##   • Level Stabilitas: Sangat Stabil
## 
## Regional Analysis:
##   • Jumlah Wilayah: 2965
##   • Wilayah Tertinggi: -4.49032 (2 hotspot)
## 
## Output Files:
##   • CSV: 5 file(s)
##   • PNG Plots: 5 file(s) (300 DPI)
##   • Interactive Map: 1 HTML file
##   • R Objects: 1 RDS file
## 
## Waktu Eksekusi: 0.5 menit
## Lokasi Output: D:/tubes-komstat/output
## 
## ================================================================================
```