

M.K. Pemrograman Web (AK-045216)

# **Konsep Pemrograman Web**

# Konsep Pemrograman Web (1)

- Pemrograman web: membuat aplikasi berbasis web
- Aplikasi berbasis web: Aplikasi yang dibuat dengan memanfaatkan mekanisme dan aplikasi yang sudah ada pada sistem web (WWW)
- Sistem web sebenarnya merupakan aplikasi yang:
  - berarsitektur client-server
    - software web browser di sisi client
    - software web server di sisi server
  - menggunakan protokol HTTP dalam komunikasi antara client dan server
  - mempunyai fungsi untuk mengambil/menjalankan isi file dokumen web di server dan menampilkannya di sisi client

# Konsep Pemrograman Web (2)

- Membuat aplikasi berbasis web berarti:
  - Memperkaya fungsi web server dengan cara menambahkan program pada dokumen web yang akan dieksekusi oleh server ketika file dokumen web tersebut diakses oleh web server
    - Misalnya, program yang mengambil data ke basis data untuk ditampilkan ke web browser
  - Memperkaya interaktivitas dokumen dengan cara menambahkan program pada dokumen web yang akan dieksekusi oleh web browser ketika file dokumen tersebut ditampilkan oleh web browser
    - Misalnya, program yang memvalidasi data masukan pada form sebelum disubmit ke web server

# Kelebihan & Kekurangan Aplikasi Berbasis Web

- Kelebihan:
  - Dapat diakses kapan pun dan dari mana pun selama ada internet
  - Dapat diakses hanya dengan menggunakan web browser (umumnya sudah tersedia di PC, PDA, dan handphone terbaru), tidak perlu menginstall aplikasi client khusus
- Kekurangan:
  - Antarmuka yang dapat dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen web dan keterbatasan kemampuan web browser untuk menampilkannya
  - Terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat

# Yang perlu dipelajari untuk membuat aplikasi berbasis web

- Di sisi client:
  - Sintaks pembuatan dokumen web (HTML & CSS)
  - Client side scripting (JavaScript)(semua ini sudah dipelajari di mata kuliah IF1191 Pemrograman Web)
- Di sisi server
  - Mekanisme pemanggilan program dan pengambilan output program oleh web server (CGI)
  - Server side scripting (PHP, JSP, ASP, dll. Untuk contoh: PHP)
- Penghubung
  - Sintaks pengalamatan dokumen web (URL)
  - Protokol komunikasi (HTTP)

# Mekanisme CGI

## (Common Gateway Interface)

- Jika dokumen web yang diminta oleh web browser merupakan file program atau file HTML yang disisipi program, maka web server akan menjalankan (*run*) file tersebut dengan bantuan interpreter atau sistem operasi
- Informasi yang diberikan oleh web server kepada program:
  - **Server variables**: berbagai informasi yang ada pada *HTTP request* yang sedang diterima, nama dan path file program, kapabilitas web server, dll
  - **Environment variables**: informasi yang telah diset (mungkin oleh aplikasi lain) pada lingkungan sistem operasi
  - **Cookie**: isi cookie yang disimpan di web browser
  - **Request parameter**: input dari user yang dikirimkan bersamaan dengan *HTTP request* oleh web browser
- Hasil output program ke *standard output (screen)* akan diambil oleh web server untuk dikirimkan kepada web browser sebagai *HTTP response*

# CGI pada PHP

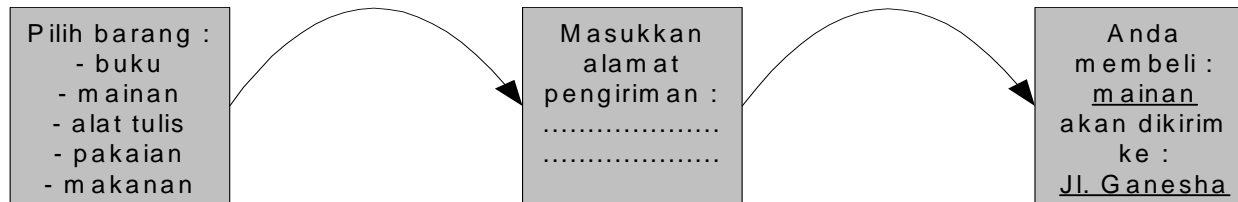
- Pada PHP:
  - Server variables diletakkan pada predefined variable `$_SERVER`
    - `$_SERVER["PHP_SELF"]` = path dan nama file yang sedang dieksekusi
    - `$_SERVER["SERVER_NAME"]` = nama host/server
    - `$_SERVER["REQUEST_METHOD"]` = jenis metode request
    - `$_SERVER["HTTP_USER_AGENT"]` = identitas web browser yang melakukan request
    - `$_SERVER["REMOTE_ADDR"]` = nomor IP user
    - dll
  - Environment variables diletakkan pada predefined variable `$_ENV`
  - Cookie diletakkan pada predefined variable `$_COOKIE`
  - Request parameter diletakkan pada predefined variable `$_GET`, `$_POST`, `$_FILES`
  - Output menggunakan perintah `echo`

Penanganan *State*



# Konsep “*task*” dalam aplikasi berbasis web

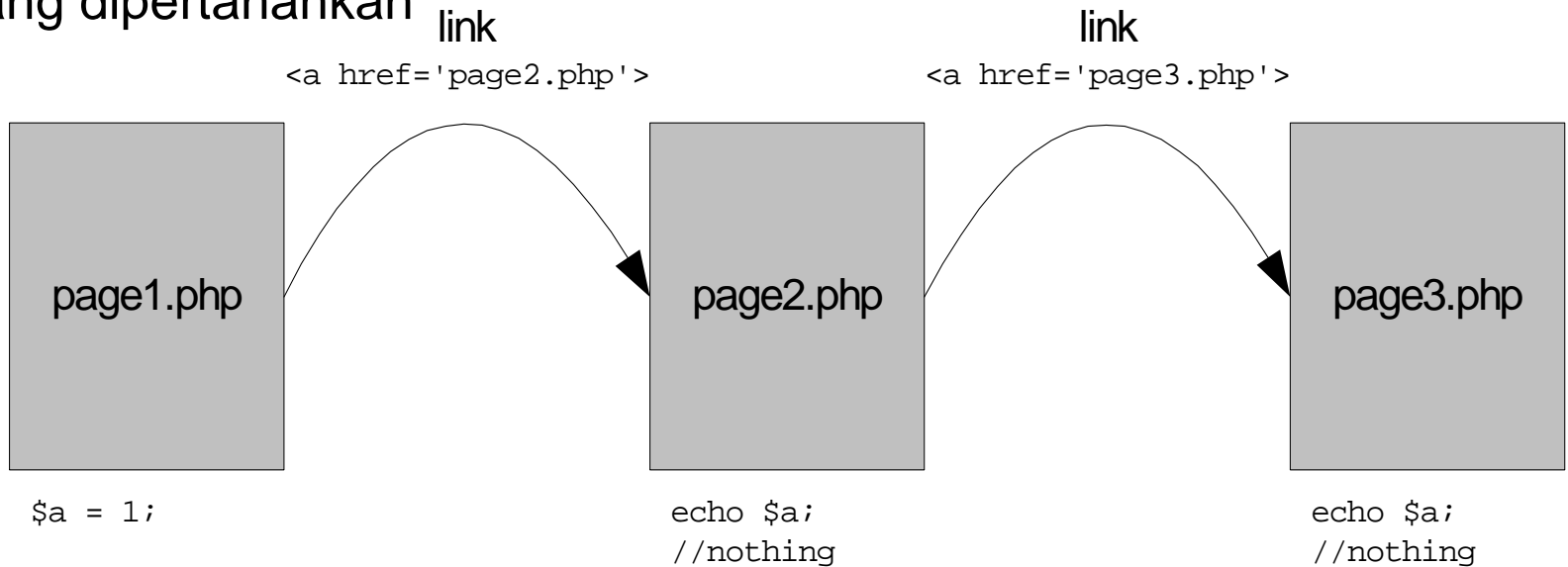
- Aplikasi mempunyai sejumlah fitur/fungsi yang memungkinkan user untuk melakukan sejumlah *task*
- Untuk memenuhi suatu *task*, dapat digunakan satu atau lebih halaman web sebagai antarmuka
- Contoh *task* yang menggunakan satu halaman web:
  - Membaca artikel nomor tertentu
  - Menambahkan entry baru pada guest book
- Contoh *task* yang menggunakan lebih dari satu halaman web:
  - Membeli barang secara online
    - halaman 1: menampilkan daftar barang yang ada untuk dipilih
    - halaman 2: mengisi alamat pengiriman
    - halaman 3: menampilkan informasi transaksi yang baru saja dilakukan



- Antara halaman web yang satu dengan yang lain harus ada komunikasi mengenai *state* dari *task* (misalnya, halaman 3 butuh informasi mengenai barang yang telah dipilih pada halaman 1)

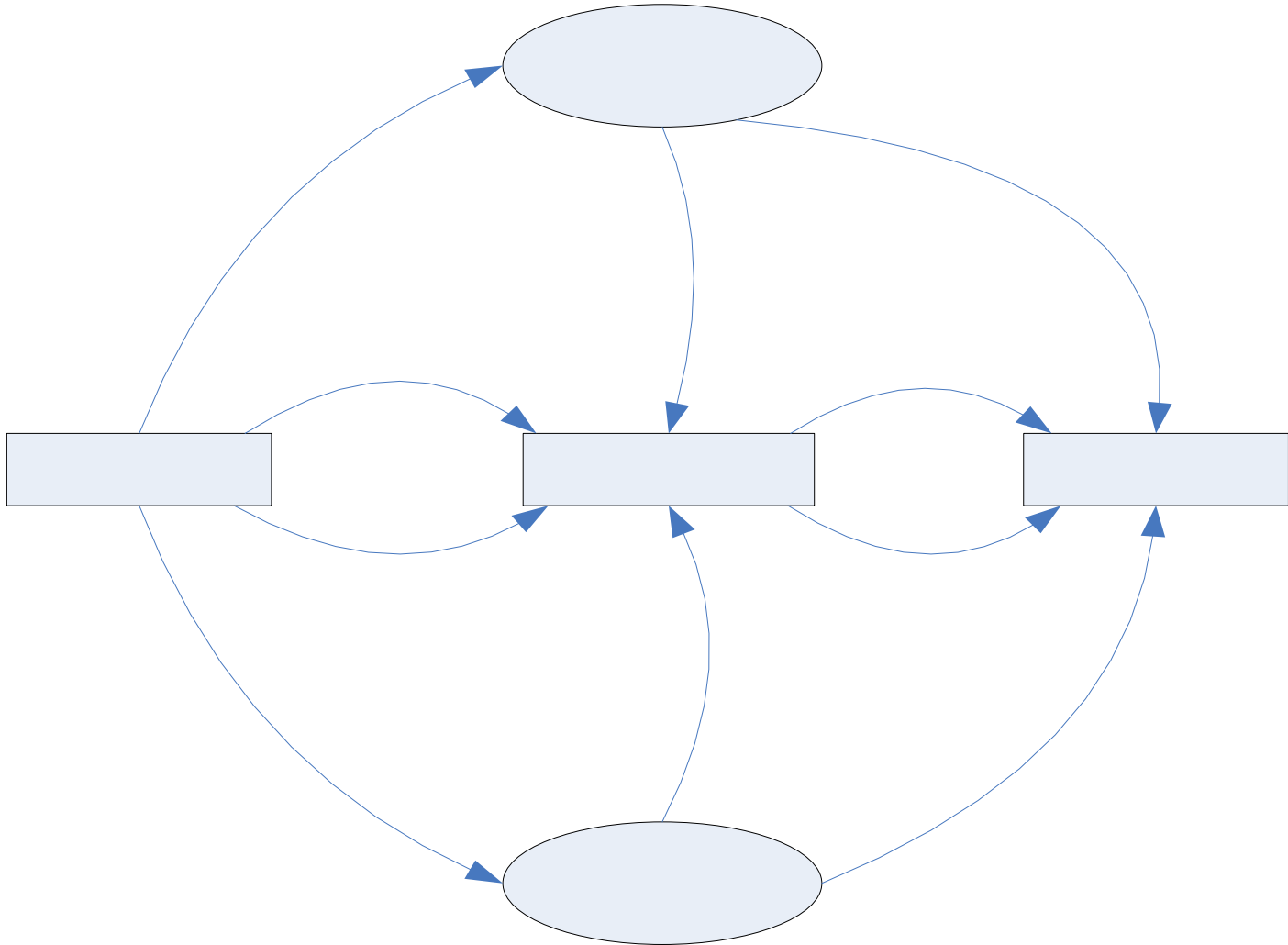
# Stateless HTTP

- Permasalahan: HTTP bersifat *stateless*: satu request saling independen terhadap request lainnya, tidak ada *state* (variabel) yang dipertahankan



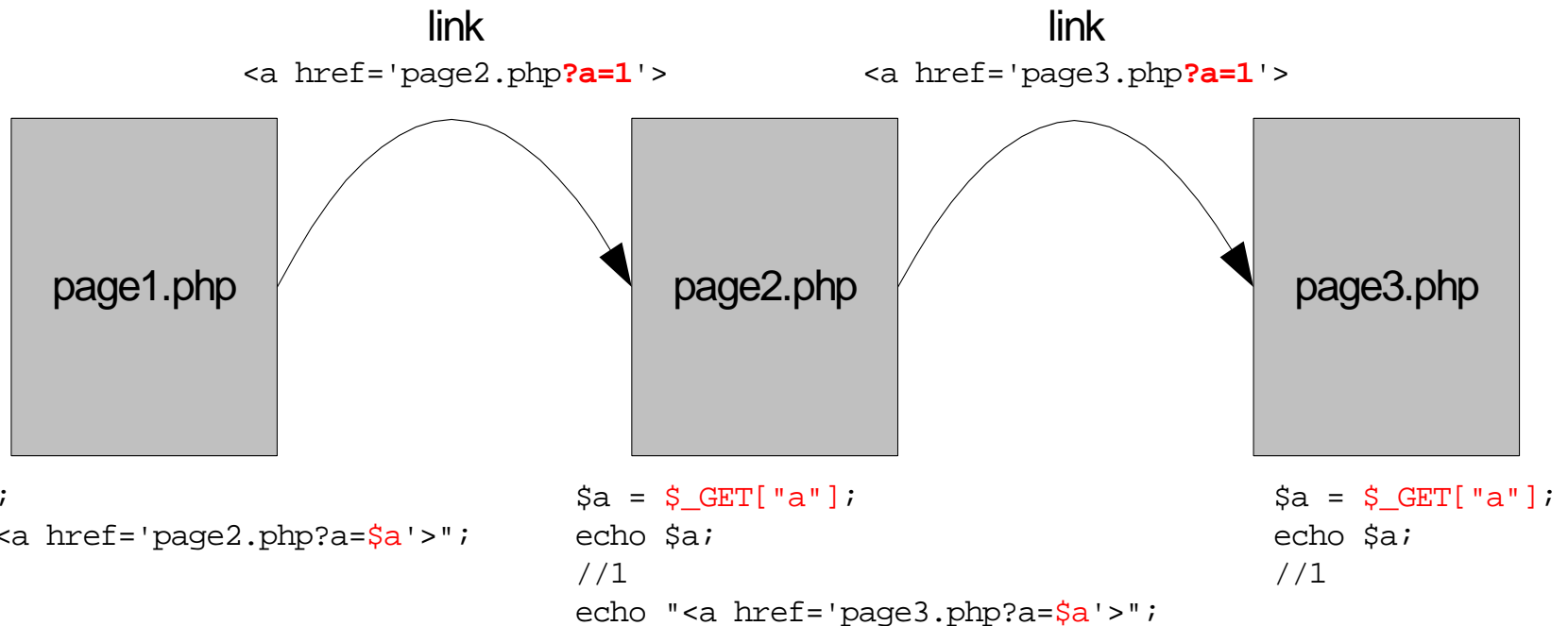
- Cara untuk mengatasi HTTP yang *stateless*:
  - Message passing via URL/Form
  - Cookie
  - Session

# Penanganan *State*



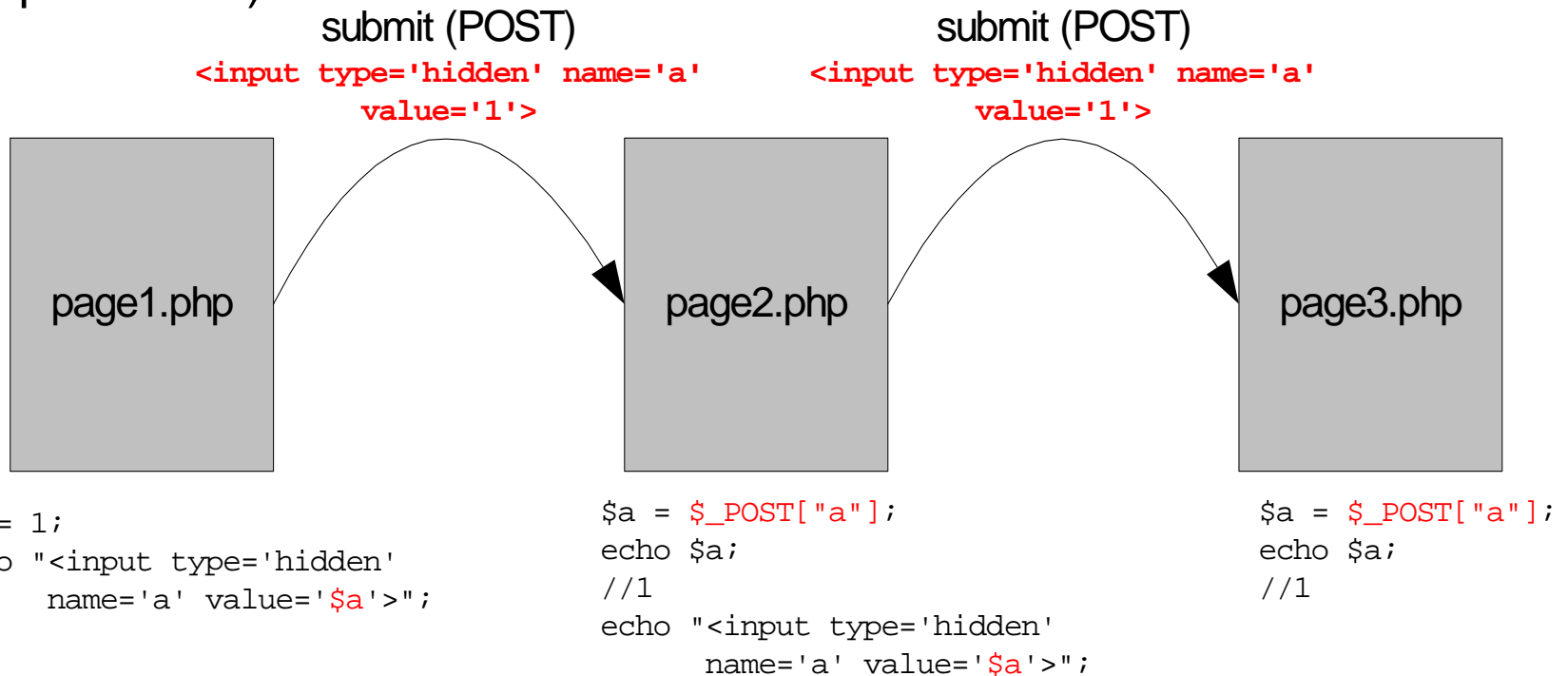
# Message Passing via URL

- Nilai variabel dilewatkan melalui parameter URL
- KIRIM: mengkonstruksi parameter pada URL
- TERIMA: akses variabel `$_GET`



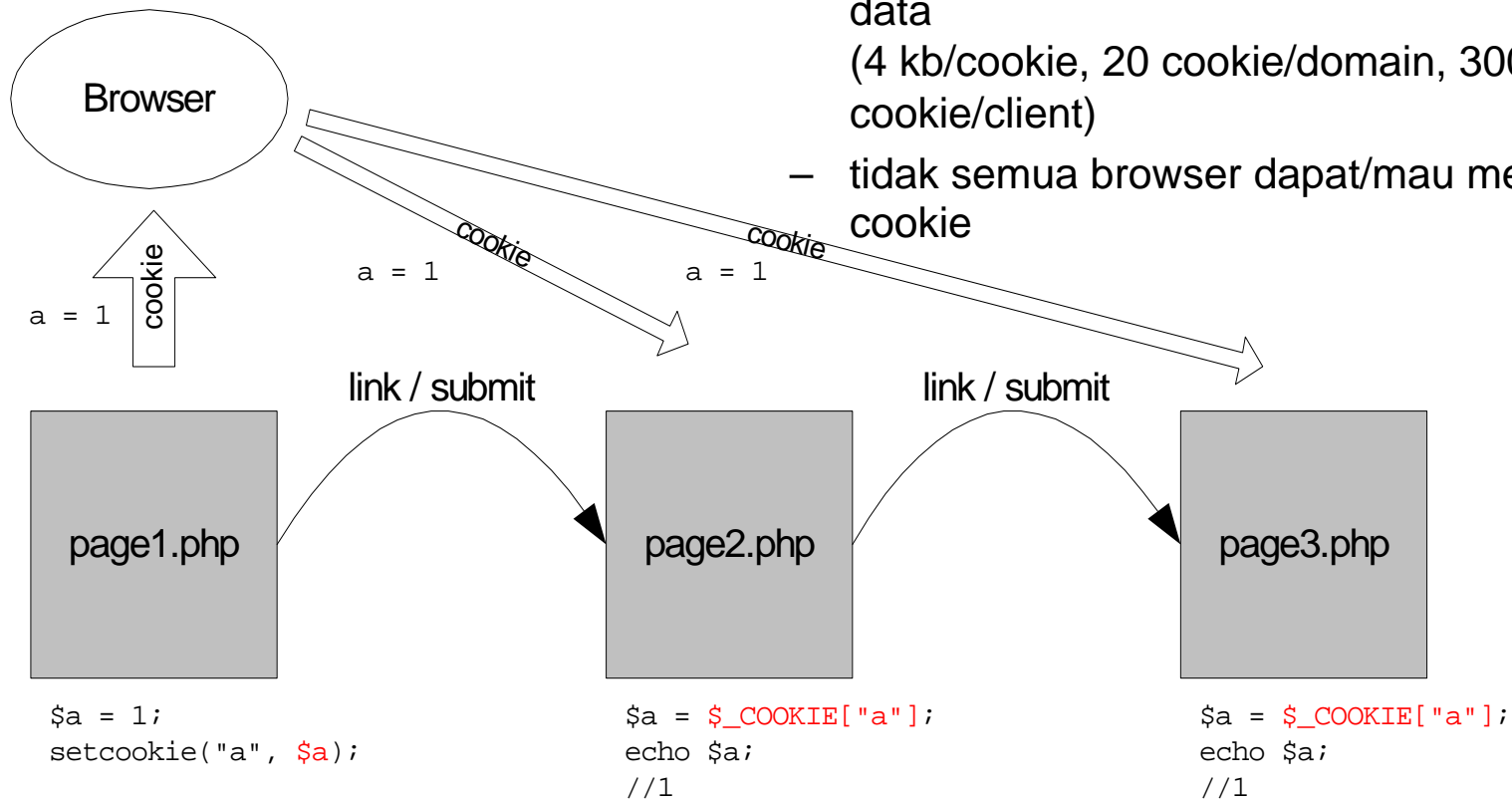
# Message Passing via Form

- Nilai variabel dilewatkan melalui input tipe hidden pada form HTML
- KIRIM: mengkonstruksi input tipe hidden dalam form
- TERIMA: akses variabel `$_GET` atau `$_POST` (tergantung method pada form)



# Cookie

- Nilai variabel disimpan di suatu tempat penyimpanan (file) di browser
  - KIRIM: `setcookie("name", "value");`
  - TERIMA: akses variabel `$_COOKIE`
  - Keterbatasan:
    - cookie hanya dapat menyimpan sejumlah kecil data  
(4 kb/cookie, 20 cookie/domain, 300 cookie/client)
    - tidak semua browser dapat/mau menerima cookie



# Session

- Nilai variabel disimpan di suatu tempat penyimpanan (file) di server
- KIRIM dan TERIMA: akses variabel `$_SESSION`

- Fungsi untuk session handling: `session_start()`, `session_id()`, `session_destroy()`, dll.
- Keterbatasan:
  - Session bergantung pada mekanisme state handling lain untuk menyimpan ID session

