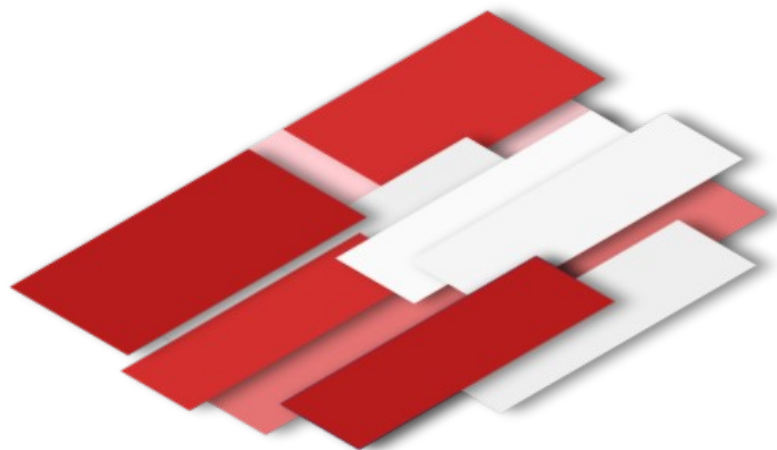


Static Site Generator

Introduction to SSG for beginner
for Personal Educational Purpose.

Jakarta, Indonesia
17 February, 2019



About Epsi



Yet, another underachiever.
But hey, I have my own blog.

SORRY FOR MY ENGLISH

About This Material

After watching this, you will understand:

How SSG can be utilized
to make a **fully mature website**
without the burden of **complicated backend**.

This material is not really comprehensive.
I still have so much to learn.

What is Static Site Generator?

What is **SSG**?



SSG short abbreviation

Static Sites Generator.



Static Sites

Just plain HTML + CSS + JS.



Static

No dynamic server (backend).



Generated

- Built once, on every change
 - Put on server
 - Serve as many request as needed.
-

What is SSG good for?



Company Profile

or Portfolio.



Personal Blog

Just like mine.



Official Site

Many Linux Distro use SSG



Any Static Sites

Such as CSS Tricks.

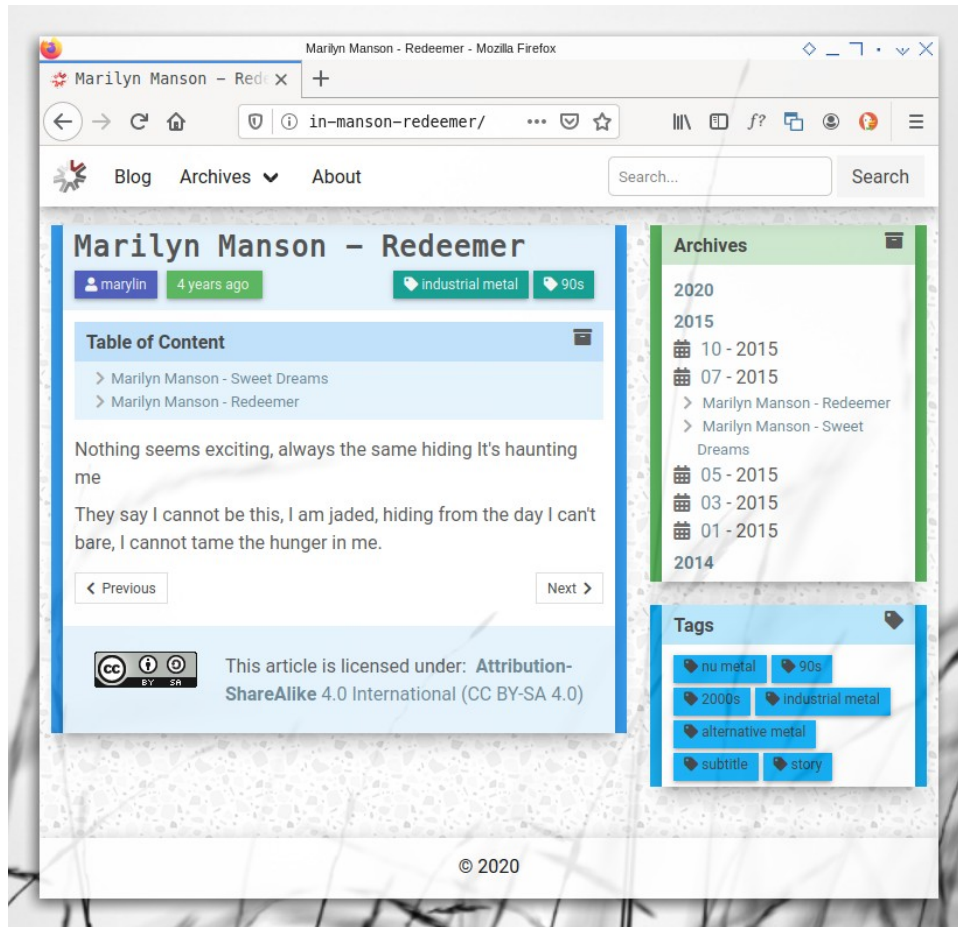


JAM Stack

Dynamic sites utilizing javascript

Modern Web Era, in contrast with LAMP stack

Blog Preview



What case is not?



Non tech savvy end user

should use blogger, or wordpress instead.



Startup

With a lot of infrastructure.



Dynamic website

Anything that require login.



Walk your dog, or feed your cat

You still have to do it yourself.

Why Static Site Generator?

Why **Static Sites**?



Markdown

More on Writing Content.



No complex backend

Focus on frontend: Great theme.



Local Copy

Own Your Own Site.



Git-able

- Cheap.
 - Free Hosting = murah meriah.
-

Tech Savvy



Embrace modern technology

SPA, JAM stack, serverless, CI/CD.



Easier to migrate content between generator

After all, it is just markdown.



OS Agnostic (simpler ecosystem)

No racial issue between, Windows, Mac and Linux/BSD user.



Easier to port tailor made custom theme

No scary language feature.



Many Option of SSG Choices

Jekyll, Hugo, Hexo, Pelican, Gatsby, 11ty.



Way cooler web developer

Over traditional CMS

Other Technical Performance



Fast

No server side processing.



Less Security Issue

SQL Injection and Vulnerabilities.



Cacheable

Can be cached reliably,
and served as CDN.



No scaling issue

It is all just static html and assets.

User Tasks

Simplified User Tasks

Only consist of this two tasks

- 1) **Design Theme and Layout**
- 2) **Writing Content, then Deploy**

No complex backend setting.

It means, more focus on the content.

Especially, with 3rd party theme.

Caveat!

Even with a ready to use theme,
you still **need customization**
to **suit your needs**.

Install

Some are easy to install.
Some are harder for non tech-savvy.

Depend on the choice of SSG.

Own Your Own Site

You can have it all locally.

- Writing content in your favorite text editor
- No upload image.
 - Edit image directly in your notebook.
- No need to dump database.

In short:

More control

Theming

Split HTML Task

- **Theme:**
 - Structure (Layout, and Skeleton)
- **Markdown Engine:**
 - Formatting

In short:

Clear division

between these two tasks.

Theme Making



Template Engine

Block and Basic Coding.



HTML Skeleton

Layout Structure.



Stylesheet

Presenting Nice Looks.

Template Engine

liquid, ejs, nunjucks, jinja2, etc...



Block Building



Basic Language

Variable, Loop, Conditional.



Static Data Files

from yaml, or toml, or json.

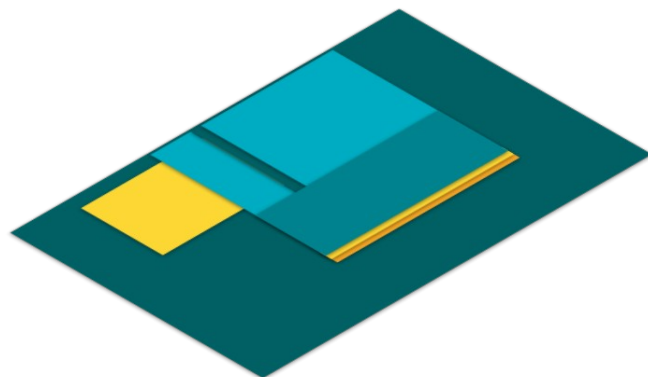


Specific Feature

Pipeline.

Stylesheet

You might also need to understand CSS Tools



[CSS Frameworks and CSS Tools]

Yet Another Presentation.

Main Configuration

Contain:

Global Stuff

Page



Frontmatter:

Per-page configuration



Content:

Markdown or HTML

Free Tutorials

Step by Step Examples?



Eleventy (11ty) Article Series

[[Eleventy Overview](#)]

(Nunjucks + Bulma MD)



Jekyll Article Series

[[Jekyll Overview](#)]

(Liquid + No Stylesheet)



Hugo Article Series

[[Hugo Summary](#)]

(Template + Bulma)



Hexo Article Series

[[Hexo Overview](#)] (EJS + Bulma)



Custom

Make your own case, and share!

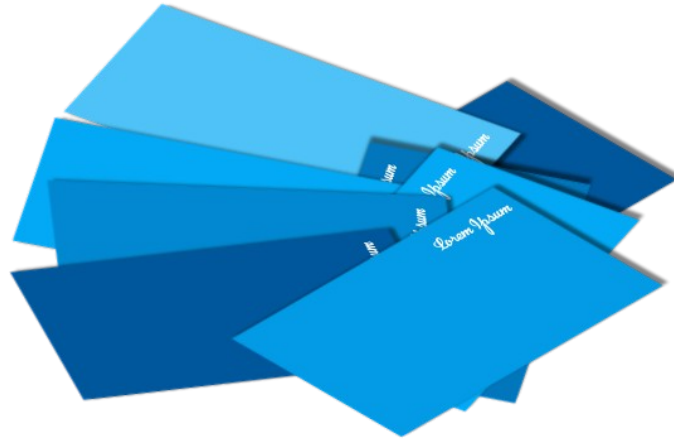
[[Hugo Summary](#)]

(Template + Bootstrap)

More examples on Repository (11ty, Jekyll, Pelican).

Markdown

Why Markdown?



More on Writing Content!

Issue with HTML Content



Time spent
on doing formatting

Wasted time = Less creativity.



Easily get lost
of concentration in writing

Dammit, my article is sucks!



Reformatting

Whenever design changes.



Error prone

Dad, I forget to close the tag again.

In short:

Markup focus on document formatting

Markdown Content

Recipe:

Markdown focus on document structure



Easier to write content

Plain Text = Painless Text



Taking care of

HTML formatting by Markdown engine

Significantly reduce formatting time while writing content.

Consistent formatting, by not doing manually.



Inline Code

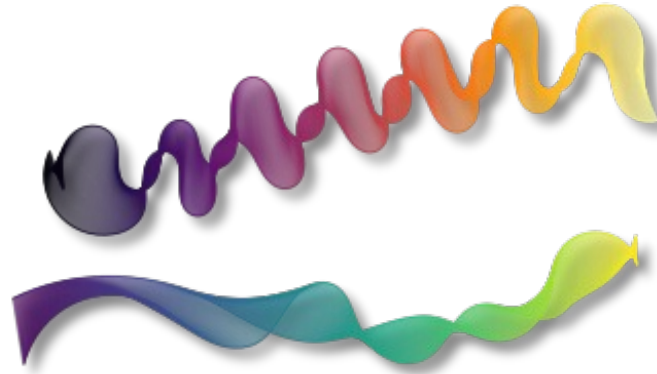
Parsed, auto format with nice color



Text editor is sufficient

No need for sophisticated tools

Markdown Engine



Implementation may vary

Each static site,
have their own markdown engine.

Be aware of slight difference.

HTML Content

When you need special formatting



HTML Page:

Such as landing page content.



Embedded Content: in Markdown Page

Such as nice looks table of content.

Repository

Track Changes



Use Git!

Friendly neighbourhood backup platform.



Time travelling is possible

OMG. I deleted my great idea last month!



No weird file naming.

My-revision-content
.rev1.final.revagain.txt



Multiple branch

while making themes enhancement

branch: Master, development, bootstrapv4,
bulma



Deploy!

Deploying Choices



Simple setup for CI/CD

Or a complex one if you wish.



Repository and Host

Gitlab, Github, Bitbucket



3rd Party CI/CD

Travis, Circle-CI, Pipelines, Gitlab-CI



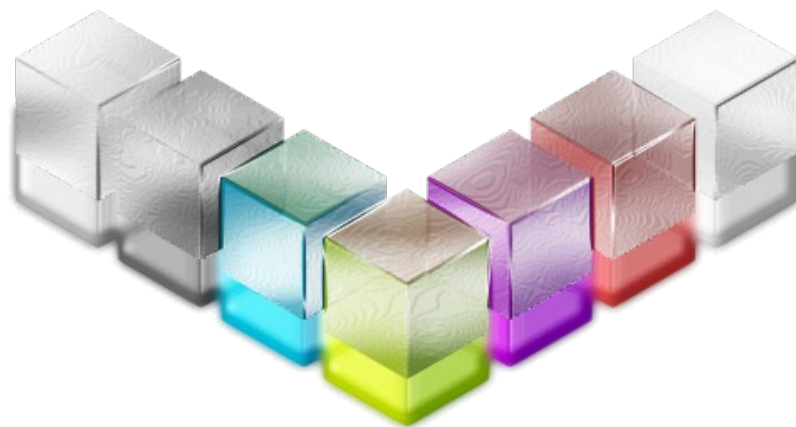
Host and CI/CD

Netlify, Vercel, Firebase

Configuration Examples

[CI/CD - Deploy Overview]

- Netlify, Zeit Now
- Manual Git: Refspec, Worktree
- Travis+Github
- CircleCI: Github, Bitbucket
- Github Workflows (Actions)



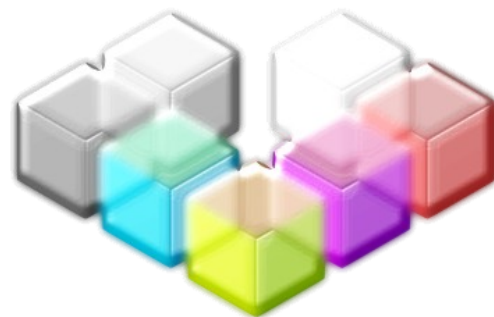
Repositories

Repositories that can directly utilize any SSG.

- github.io, and gitlab.io,

Repositories that require 3rd party.

- bitbucket.io.



Issues!

Issues with **Static Site**?



No commenting system

Except embedded



No internal search



Local build time

For site with many article, or
theme design with poor logic.

Again, Free Tutorials?

Step by Step Examples?

Revisited in Repository Fashioned



Hugo Repository

[[Hugo + Template + Bulma MD](#)]

[[Hugo + Template + Bootstrap](#)]



Jekyll Repository

[[Jekyll + Liquid + Bulma MD](#)]



Pelican Repository

[[11ty + Nunjucks + Materialize](#)]

[[11ty + Nunjucks + Bulma MD](#)]



Pelican Repository

[[Pelican + Jinja2 + Bulma MD](#)]



Hexo Repository

[[Hexo + EJS + Bulma](#)]

In Need of a Technical Material?



Eleventy

[\[Yet Another Presentation\]](#)



Jekyll

[\[Yet Another Presentation\]](#)



Hexo

[\[Yet Another Presentation\]](#)

Summary

Summary?

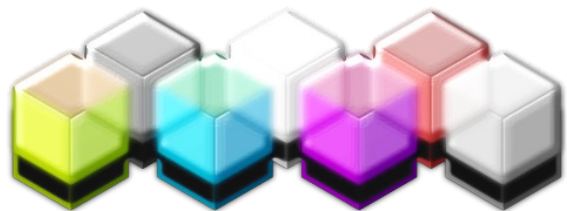
Along with local communities.

Static Site Generator:

- [[Hugo](#)], [[Jekyll](#)], Pelican,
- [[Eleventy \(11ty\)](#)], [[Hexo](#)], [[Gatsby](#)].

Template Engine:

- Nunjucks, Jinja2, EJS, Liquid,
- Local Community: [[JAMstackID](#)]



Text:

- Markdown, reStructured Text.

Deploy:

- CVS: Git, Mercurial: [[@freekelasgithub](#)],
- Repository: Gitlab, Github, Bitbucket,
- CI/CD: Travis, Circle-CI, Pipelines: [[IDDevOps](#)],
- Static Host: Netlify, Render, Zeit Now.

Looking for the right tools?

A real world example require complete stack.

Do Your Own Research!



[Practical Blog Development Cycle]

Learning The Hard Way.



What is Next?

Leverage to JAM stack!

[staticgen.com]

[[@JAMstackID](#)]

Questions?

Don't be shy!



The End

Thank You for Your Time.