

Pedoman Pengguna Drupal 8



Drupal™



USER GUIDE

Pedoman Pengguna Drupal 8

Daftar Isi

i. Pengantar	vi
i.1. Copyright	vi
i.2. Audience and Goal	vi
i.3. Organization	vii
i.4. Reporting Problems	viii
i.5. Conventions of the Guide	ix
i.6. Guiding Scenario	xi
1. Memahami Drupal	1
1.1. Concept: Drupal as a Content Management System	1
1.2. Concept: Modules	3
1.3. Concept: Themes	4
1.4. Concept: Distributions	5
1.5. Concept: Types of Data	6
1.6. Concept: The Drupal Project	6
1.7. Concept: Drupal Licensing	8
2. Merencanakan Situs Anda	10
2.1. Concept: Regions in a Theme	10
2.2. Planning Your Site Layout	11
2.3. Concept: Content Entities and Fields	13
2.4. Concept: Modular Content	15
2.5. Planning your Content Structure	16
2.6. Concept: Editorial Workflow	18
3. Instalasi	20
3.1. Concept: Server Requirements	20
3.2. Concept: Additional Tools	21
3.3. Preparing to Install	23
3.4. Running the Installer	26
4. Konfigurasi Dasar Situs	32
4.1. Concept: Administrative Overview	32
4.2. Editing Basic Site Information	35
4.3. Installing a Module	37
4.4. Uninstalling Unused Modules	39
4.5. Configuring User Account Settings	41
4.6. Configuring the Theme	43
5. Manajemen Dasar Halaman	47
5.1. Concept: Paths, Aliases, and URLs	47
5.2. Creating a Content Item	48
5.3. Editing a Content Item	50
5.4. Editing with the In-Place Editor	52
5.5. Designating a Front Page for your Site	53
5.6. Concept: Menu	54
5.7. Adding a Page to the Navigation	56
5.8. Changing the Order of Navigation	58
6. Menyiapkan Struktur Konten	61
6.1. Adding a Content Type	61
6.2. Deleting a Content Type	65

6.3. Adding Basic Fields to a Content Type	66
6.4. Concept: Reference Fields	71
6.5. Concept: Taxonomy	72
6.6. Setting Up a Taxonomy	73
6.7. Adding a Reference Field	78
6.8. Concept: Forms and Widgets	82
6.9. Changing Content Entry Forms	82
6.10. Concept: View Modes and Formatters	85
6.11. Changing Content Display	86
6.12. Concept: Image Styles	89
6.13. Setting Up an Image Style	90
6.14. Concept: Responsive Image Styles	93
6.15. Concept: Text Formats and Editors	94
6.16. Configuring Text Formats and Editors	95
7. Mengelola Akun Pengguna	98
7.1. Concept: Users, Roles, and Permissions	98
7.2. Concept: The User 1 Account	99
7.3. Creating a Role	100
7.4. Creating a User Account	102
7.5. Assigning Permissions to a Role	105
7.6. Changing a User's Roles	107
7.7. Assigning Authors to Content	109
8. Blok	112
8.1. Concept: Blocks	112
8.2. Creating A Custom Block	112
8.3. Placing a Block in a Region	114
9. Membuat Daftar dengan Views	117
9.1. Concept: Uses of Views	117
9.2. Concept: The Parts of a View	118
9.3. Creating a Content List View	119
9.4. Duplicating a View	124
9.5. Adding a Block Display to a View	127
10. Membuat Situs Anda Multilingual	132
10.1. Concept: User Interface, Configuration, and Content translation	132
10.2. Adding a Language	133
10.3. Configuring Content Translation	134
10.4. Translating Content	138
10.5. Translating Configuration	139
11. Mengembangkan dan Menyesuaikan Situs Anda	143
11.1. Finding Modules	143
11.2. Enabling and Disabling Maintenance Mode	147
11.3. Downloading and Installing a Module from <i>Drupal.org</i>	149
11.4. Finding Themes	152
11.5. Downloading and Installing a Theme from <i>Drupal.org</i>	155
11.6. Manually Downloading Module or Theme Files	158
11.7. Concept: Development Sites	160
11.8. Making a Development Site	161
11.9. Using Composer and Git to Download Files	163
11.10. Deploying New Site Features	166

11.11. Synchronizing Configuration Versions	169
12. Mencegah dan Memperbaiki Masalah	172
12.1. Concept: Cache	172
12.2. Clearing the Cache	173
12.3. Concept: Data Backups	174
12.4. Concept: Log	176
12.5. Concept: Status Report	177
13. Keamanan dan Pemeliharaan	179
13.1. Concept: Cron	179
13.2. Configuring Cron Maintenance Tasks	179
13.3. Concept: Security and Regular Updates	182
13.4. Keeping Track of Updates	183
13.5. Updating the Core Software	184
13.6. Updating a Module	186
13.7. Updating a Theme	188
14. Pemikiran Akhir	192
14.1. Connecting with the Community	192
14.2. Getting Support	193
14.3. Learning More	195
Glosarium	196
Indeks	202
A. Lampiran	212
A.1. Guide-Wide Attributions	212

Pengantar

i.1. Copyright

This guide was written by contributors to the Drupal open-source project. It is copyright 2015-2016 by the individual contributors, and can be used in accordance with the [Creative Commons License, Attribution-ShareAlike 2.0](#) (CC BY-SA 2.0). Each page in this document (including this one) has an Attributions statement at the bottom, listing the people who contributed to writing and editing that page. See also [Bagian A.1, "Guide-Wide Attributions"](#) for guide-wide editing, translating, and project management information.

The CC BY-SA license (very similar to the GPL) allows anyone to copy, modify, and redistribute modifications of all or part of this work, as long as the following is complied with:

- You provide appropriate credit (see the license for more information).
- You provide a link to the license.
- You indicate whether changes have been made.
- You distribute your work under the same license as this original.

Attributions

This page was adapted and edited by [Jennifer Hodgdon](#), and [Jojo Alphonso](#) at [Red Crackle](#), from ["Documentation copyright and licensing"](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

Copyright notice: Copyright 2015-2016 by the individual contributors; see [Bagian i.1, "Copyright"](#) for details. Licensed under [CC BY-SA 2.0](#).

i.2. Audience and Goal

This guide was written mainly for people with minimal knowledge of the Drupal content management system. The topics will help them become skilled at installing, administering, site building, and/or maintaining the content of a Drupal-based website. The guide is also aimed at people who already have some experience with a current or past version of Drupal, and want to expand the range of their skills and knowledge or update them to the current version.

This guide assumes that you have already decided you want to learn and use Drupal. If you need to learn more before deciding, see [Bagian 1.1, "Concept: Drupal as a Content Management System"](#).

Depending on which aspects of Drupal you would like to learn, you will need some background knowledge to understand this guide: general Internet skills and knowledge are assu-

med, and the guide concentrates on how to use the software itself. For instance, the sections about installing Drupal on a web server assume you can obtain web hosting and figure out how to transfer files to your chosen web host. Similarly, the sections about content management assume you can log into a website and fill in a web-based form.

After reading this guide, you should be able to:

- Plan the content architecture for a Drupal-based site
- Build the site that you planned
- Manage and administer your site
- Understand documentation and blog posts on topics not covered here, to expand your knowledge and skills
- Connect with the worldwide Drupal community

Attributions

Written by [Jennifer Hodgdon](#).

i.3. Organization

This user guide contains a series of topics, each of which covers either a *task* (how to do something) or a *concept* (background knowledge, terminology, and the like). Concept topics have names starting with *Concept*:; while task topics have names containing verbs, like *Editing Basic Site Information*.

The topics are grouped into chapters in a logical order, with concepts and tasks interleaved so that concepts are presented before related tasks, and tasks build on each other. To take advantage of this, you are encouraged to read the entire guide in its presented order, possibly skipping topics that are not of interest or that present information you already know. Remember to try out the tasks on your own site as you read the guide; most people learn better by doing rather than reading.

If you prefer, you can also use the index or table of contents to jump straight to a topic that you'd like to learn about, rather than reading the entire guide. To facilitate this approach, each topic lists the prerequisite knowledge that you'd need in order to understand it, if any (with links to the topics that present that knowledge); task topics also list *site prerequisites* (things that you would need to have configured or created on your site in order to perform the task). Also, most topics have sections at the end where you can find related information and/or tasks for expanded understanding, to continue your learning.

You may also want to refer to the [Glosarium](#) section as you read — it gives brief definitions of most of the terminology used in the guide, with links to topics having more detailed explanations.

Attributions

Written by [Jennifer Hodgdon](#).

i.4. Reporting Problems

Goal

Report a problem with this guide, such as:

- Information that is incorrect or does not follow best practices
- Steps that do not work
- Screen shots or text that doesn't match what you see on the screen
- Unclear writing
- Places where a table or screen shot would help clarify the text
- Failure to define terminology
- Missing knowledge prerequisites or site prerequisites for a topic
- Typographical, spelling, grammar, or formatting errors
- Broken links

Steps

1. Make a note of the topic or topics that contain the problem you have found.
2. Log in to [Drupal.org](#) (you will need to create a user account if you do not already have one).
3. Visit the [User Guide issues page on Drupal.org](#).
4. Verify that the problem you found has not already been reported in another issue:
 - If there are only a few open issues, scan the *Summary* column to see if any of their descriptions match the problem you found. You may also need to read some of the issues to make sure, which you can do by clicking the links in the *Summary* column.
 - If the open issue list is long, enter either a keyword related to the problem you found or the title of the topic where the problem occurs in the *Search for* box, and click *Search* to reduce the issue list. Then either scan the summaries or read the issues to see if they match your problem.
5. If you determine that your problem has not already been reported, click *Create a new issue*, and fill in the issue report as follows:

Field name	Explanation	Example value
Title	Short summary of the problem you found	Instructions in "Adding a Content Type" do not work
Category	Type of issue being reported	Bug report

Field name	Explanation	Example value
Version	Version of the guide you found the problem in	8.x-0.x-dev
Issue summary	Details of the problem you found	In the "Adding a Content Type" topic, in step 3, when I clicked <i>Save</i> , I got the following error message: ...

6. Reread the *Title* and *Issue summary* you entered, and verify that the following information is included in your report:
 - A complete description of the problem you found
 - The name of the topic or topics where you found the problem
 - The language you are reading the guide in (if not English)
 - If you read the guide on a website, a link to the page or pages with the problem
7. Click *Save* to create the issue.
8. Check back on the issue in a few days. If one of the project maintainers has asked for clarification, respond by adding a comment to the issue.

Attributions

Written by [Jennifer Hodgdon](#).

i.5. Conventions of the Guide

Assumptions and prerequisites

This guide has the following assumptions and prerequisites:

- This guide is organized into topics; see [Bagian i.3, "Organization"](#) for details. Many topics include a *Prerequisite knowledge* section, which lists other topics whose content knowledge is needed in order to understand the topic you are reading. Some background knowledge that is not covered in the guide is also assumed; see [Bagian i.2, "Audience and Goal"](#) for details.
- Many task topics list *Site prerequisites*, which are tasks that you'll need to have completed on your site before you'll be able to do the task in the topic you are reading.
- The specifics of the site prerequisites relate to the scenario used throughout this guide of building a site for a farmers market (see [Bagian i.6, "Guiding Scenario"](#) for details). You can adapt the tasks to your own scenario, but you will also need to remember the changes you made when deciding if your site satisfies the site prerequisites for a task.
- For all task topics after [Bagian 3.4, "Running the Installer"](#), there is also an implicit prerequisite: you must have installed the content management software on your site, and be logged in to a user account with sufficient permissions to do the task (such as the user account created when you installed your site, which automatically has full permissions).

- If you read all the topics in order, and perform all of the steps in the task topics as you go (staying logged in), you should have the background knowledge and site prerequisites in place for each topic as you read it.

Text conventions

The following conventions are used in the text of this guide:

- The URL *example.com* means the base URL of your website. See the Navigation section below for more details on how URLs internal to your site are indicated.
- Text you should see in the user interface of your site is shown in *italics*, such as: Click *Save configuration*. This only applies to text in the user interface that comes from the software, not to text that was entered in a previous topic. For example, in a topic about editing, you might see this instruction: Click *Edit* in the row of the About page (*Edit* would be in *italics*, but *About* would not be, because the About page was created in a previous topic).
- URLs, file names, and newly-introduced terminology are also shown in *italics*.
- Text that you should type at a shell command line is shown in monospace type, such as:

```
drush cr
```
- Within this guide, the word *directory* is always used to refer to file directories (which some people prefer to call *folders*).

Navigation

To do most of the task topics in this guide, you will need to navigate to one or more pages in the administrative interface of your site. You might see something like this in the instructions (this will make more sense after you have the base software installed):

In the *Manage* administrative menu, navigate to *Structure > Taxonomy* ([admin/structure/taxonomy](http://example.com/admin/structure/taxonomy)).

Navigation instructions like this assume that you have the core Toolbar module installed, and this example means that in the menu bar at the top of your site, you would need to click *Manage* to expose the menu choices, then click *Structure*, then *:_Taxonomy_*, and that at the end, you would be on a page with URL <http://example.com/admin/structure/taxonomy> (if your site base URL is <http://example.com>).



Here's another example:

In the *Manage* administrative menu, navigate to *Configuration > System > Site information* ([admin/config/system/site-information](http://example.com/admin/config/system/site-information)).

In this example, after clicking on *Manage* and *Configuration*, you would need to find the *System* section of the page, and within that, click *Site information*. After that, you'd end up on <http://example.com/admin/config/system/site-information>.

SYSTEM

Basic site settings
Change site name, email address, slogan, default front page, and error pages.

Cron
Manage automatic site maintenance tasks.

One other note: if you are using the standard administrative core Seven theme, many "Add" buttons in the administrative interface are displayed with + signs on them. For instance, on admin/content, the Add new content button appears as + *Add new content*. However, this is theme-dependent and is not really part of the text on the button (for instance, it would not necessarily be read by a screen reader), so in this guide, the convention is to not mention the + sign on the buttons.

Filling in forms

Many of the task topics in this guide include steps where you will fill out a web form. In most cases, a screen capture image of the form will be included, along with a table of the values you will need to enter into each form field. For example, you might see a table that starts out like this, explaining the site information form you would see if you navigated to *Configuration > System > Site information* (admin/config/system/site-information):

Field name	Explanation	Example value
Site name	Name of your site	Anytown Farmers Market

To use this table, find the field labeled *Site name* in the form, and enter the name of your site in this field. An example site name of "Anytown Farmers Market" is suggested in the table, which relates to the scenario of building a website for a farmers market that you'll find all through this guide (see [Bagian i.6, "Guiding Scenario"](#) for details).

Attributions

Written/edited by [Jennifer Hodgdon](#).

i.6. Guiding Scenario

When reading this guide, it is helpful to have a website building project in mind. The following project scenario provides context and links together the examples in this guide:

You are making a website for a farmers market. The site needs to display information about the location and hours of the market, and an About page with the history of the market.

It also needs to list the vendors. Vendors should be able to edit their listings (including a logo or photo), and post recipes. Site visitors should be able to browse recipes, or locate recipes using ingredients that they purchased at the market. Some visitors to your site speak another language, so the main pages and vendor pages need to be translated.

Attributions

Written/edited by [Jennifer Hodgdon](#).

Bab 1. Memahami Drupal

1.1. Concept: Drupal as a Content Management System

What is a Content Management System?

A content management system (CMS) is a software tool that lets users add, publish, edit, or remove content from a website, using a web browser on a smartphone, tablet, or desktop computer. Typically, the CMS software is written in a scripting language, and its scripts run on a computer where a database and a web server are installed. The content and settings for the website are usually stored in a database, and for each page request that comes to the web server, the scripts combine information from the database and *assets* (JavaScript files, CSS files, image files, etc. that are part of the CMS or have been uploaded) to build the pages of the website.

The combination of the operating system that the CMS runs on, the scripting language it is written in, the database it stores its information in, and the web server that runs the scripts to retrieve information and return it to the site visitor's web browser is known as the *stack* that the CMS runs on; the commonly used combination of the Linux operating system, Apache web server, MySQL database, and PHP scripting language is known as the *LAMP stack*.

What is Drupal?

Drupal is a flexible CMS based on the LAMP stack, with a modular design allowing features to be added and removed by installing and uninstalling *modules*, and allowing the entire look and feel of the website to be changed by installing and uninstalling *themes*. The base Drupal download, known as Drupal Core, contains the PHP scripts needed to run the basic CMS functionality, several optional modules and themes, and many JavaScript, CSS, and image assets. Many additional modules and themes can be downloaded from the [Drupal.org](https://www.drupal.org) website.

Drupal can also run on other technology stacks:

- The operating system can be Windows or Mac OS instead of Linux.
- The web server can be Nginx or IIS instead of Apache.
- The database can be PostgreSQL or SQLite instead of MySQL, or a MySQL-compatible replacement such as MariaDB or Percona.

Other operating systems, web servers, and databases can also be made to work; however, the scripts that the software uses are written in PHP, so that cannot be changed.

What are the reasons for using Drupal?

When building a website, you have your choice of using one of the many existing CMS packages and hosted services, developing your own CMS, or building the site without using a CMS. Here are some of the reasons you might choose to use Drupal:

- Building a small, simple site with static HTML pages is not difficult, and you can get a simple site up very quickly. Setting up a site in a CMS generally requires more time initially, but brings you the benefits of on-line editing (easier for less experienced content maintainers), uniformity (harder to maintain using static HTML for larger sites), and the possibility of more complex features requiring a database.
- Some CMS software is special-purpose; for instance, there are packages and hosted services that you can use to build a blog or a club membership website. Drupal, in contrast, is a general-purpose CMS. If you are building a special-purpose site, you might choose to use a special-purpose CMS; however, if your site falls even slightly outside the intended purpose, you will probably be better off using a general-purpose CMS rather than trying to adapt a special-purpose CMS.
- Building your own CMS-type software can seem attractive. However, using a general-purpose CMS like Drupal as a starting point is usually a better idea, because the basic CMS functionality (such as user accounts and content management) has thousands of developer hours behind it, including many years of user testing, bug fixing, and security hardening.
- Some CMS software packages are expensive to purchase a license for. Some are free or have a free version, but have restrictive licenses that do not allow you to make modifications and extensions. You might prefer to use a package (like Drupal) that has a less restrictive software license, and is developed by a world-wide community. See [Bagian 1.6, "Concept: The Drupal Project"](#) for more on this topic.

Related topics

- [Bagian 1.2, "Concept: Modules"](#)
- [Bagian 1.3, "Concept: Themes"](#)
- [Bagian 1.4, "Concept: Distributions"](#)
- [Bagian 1.6, "Concept: The Drupal Project"](#)

Additional resources

- [The Drupal overview](#)
- [Drupal.org page "FAQ" \(Frequently Asked Questions\)](#)
- [Drupal.org page "Case Studies"](#)
- [Wikipedia page "Content management systems"](#)
- [Wikipedia page "Modular design"](#)

Attributions

Written and edited by [Kristof van Tomme](#) at [Pronovix](#), [Jennifer Hodgdon](#), and [Michael Lenahan](#) at [erdfisch](#).

1.2. Concept: Modules

Prerequisite knowledge

[Bagian 1.1, "Concept: Drupal as a Content Management System"](#)

What is a module?

A module is a set of PHP, JavaScript, and/or CSS files that extends site features and adds functionality. You can turn the features and functionality on by *installing* the module, and you can turn it off by *uninstalling* the module; before uninstalling, you may need to remove data and configuration related to the feature or functionality. Each module that is installed adds to the time needed to generate pages on your site, so it is a good idea to uninstall modules that are not needed.

The core download provides modules for functionality such as:

- Managing user accounts (the core User module)
- Managing basic content (the core Node module) and fields (the core Field and Field UI modules; there are also core modules providing field types)
- Managing navigation menus (the core Menu UI module)
- Making lists, grids, and blocks from existing content (the core Views and Views UI modules)

You can download additional *contributed modules* from the [Drupal.org Module Downloads](#), or create your own *custom modules*.

Related topics

- [Bagian 1.3, "Concept: Themes"](#)
- [Bagian 1.4, "Concept: Distributions"](#)
- [Bagian 4.4, "Uninstalling Unused Modules"](#)
- [Bagian 11.1, "Finding Modules"](#)
- [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#)
- [Bagian 13.3, "Concept: Security and Regular Updates"](#)
- [Bagian 13.6, "Updating a Module"](#)

Additional resources

[Drupal.org community documentation page "Module developer's guide"](#)

Attributions

Written by [Jennifer Hodgdon](#).

1.3. Concept: Themes

Prerequisite knowledge

[Bagian 1.1, "Concept: Drupal as a Content Management System"](#)

What is a Theme?

A theme is a set of files that define the visual look and feel of your site. The core software and modules that run on your site determine which *content* (including HTML text and other data stored in the database, uploaded images, and any other asset files) is displayed on the pages of your site. The theme determines the HTML markup and CSS styling that wraps the content.

The core software provides several basic themes with the core distribution. These themes have largely been designed and built by the community over the last several years and will all be good choices for building your first sites and becoming more familiar with how the core software works.

Drupal is a well-established CMS so the market for 3rd party themes - both free and paid - is very robust.

If none of the 3rd party options suit your needs, you'll need to create a custom theme. A custom theme can be as simple as a single CSS file that adds styling to the markup provided by the core software. Guidance for creating custom themes in Drupal 8 can be found in the [Drupal.org community documentation page "Theming Drupal"](#).

Related topics

- [Bagian 11.4, "Finding Themes"](#)
- [Bagian 11.5, "Downloading and Installing a Theme from Drupal.org"](#)
- [Bagian 1.2, "Concept: Modules"](#)

Additional resources

[Drupal.org community documentation page "Theming Drupal"](#)

Attributions

Written and edited by [John Grubb](#) and [Jennifer Hodgdon](#).

1.4. Concept: Distributions

Prerequisite knowledge

- [Bagian 1.1, "Concept: Drupal as a Content Management System"](#)
- [Bagian 1.2, "Concept: Modules"](#)
- [Bagian 1.3, "Concept: Themes"](#)

What are Distributions?

Distributions provide site features and functions for a specific type of site as a single download containing the core software, contributed modules, themes, and pre-defined configuration. A distribution makes it possible to set up a complex, use-specific site in fewer steps than installing and configuring elements individually.

There are two main types of distributions:

Full-featured Distributions

A full-featured distribution is a project that provides a complete solution to set up a site for a specialized purpose such as academic, business, government, nonprofit, publishing, social, etc. For example, you could use an existing distribution for farmers markets to build your own website, or you could share your set-up for the farmers market site as a distribution for others to use.

Other Distributions

Distributions can also be quick-start tools that developers and site builders can use as a starting point.

Related topics

[Bagian 14.1, "Connecting with the Community"](#)

Additional resources

[Drupal.org Drupal Distribution Downloads](#)

Attributions

Adapted and edited by [Diána Lakatos](#) and [Antje Lorch](#), from "[Distributions](#)" and "[Download & Extend — Distributions](#)" copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

1.5. Concept: Types of Data

What are the types of data?

The data and information on your site is divided up into four types, which are edited, translated, and stored differently. These four types are:

Content

Information (text, images, etc.) meant to be displayed to site visitors. This type of information tends to be relatively permanent, but can normally be edited.

Configuration

Information about your site that is not content, but is also relatively permanent, and is used to define how your site behaves or is displayed. It is sometimes also displayed to site visitors, but tends to be smaller pieces of text (like field labels, the name of your site, etc.) rather than larger chunks that you'd normally think of as Content.

State

Information of a temporary nature about the current state of your site, such as the time when cron jobs were last run.

Session

Information about individual site visitors' interactions with the site, such as whether they are logged in and their cookies. This is technically a subtype of State information, since it is also temporary.

Related topics

- [*Bagian 2.3, "Concept: Content Entities and Fields"*](#)
- [*Bagian 13.1, "Concept: Cron"*](#)
- [*Bagian 7.1, "Concept: Users, Roles, and Permissions"*](#)
- [*Bagian 10.1, "Concept: User Interface, Configuration, and Content translation"*](#)

Attributions

Adapted by [*Jennifer Hodgdon*](#) from [*"Overview of Configuration \(vs. other types of information\)"*](#), copyright 2000-2016 by the individual contributors to the [*Drupal Community Documentation*](#)

1.6. Concept: The Drupal Project

Prerequisite knowledge

[*Bagian 1.1, "Concept: Drupal as a Content Management System"*](#)

What is Free and Open Source Software?

Free and Open Source Software (FOSS) is software that is developed by a community of people, released under a non-commercial license, and whose *source code* (the program files that make up the software) is freely available. For more information on the non-commercial license used by Drupal, see [Bagian 1.7, "Concept: Drupal Licensing"](#).

What is the Drupal project?

The Drupal project is a FOSS project whose purpose is to develop the core content management system software, as well as add-on modules, additional themes, translations, documentation, and special-purpose distributions. The people who contribute their time and money to the Drupal project come from all over the world, and are a diverse community that comes together for this common purpose.

The community encompasses many smaller groups who perform many different tasks such as developing a particular piece of Drupal-related software, writing documentation, maintaining the security of Drupal software, translating Drupal software into a particular language, using Drupal for some specific purpose, and coming together to meet in person within a particular geographical area.

For more on how you can connect to and communicate with the world-wide community, see [Bagian 14.1, "Connecting with the Community"](#) and [Bagian 14.2, "Getting Support"](#).

What is the Drupal Association?

The [Drupal Association](#) is a non-profit organization dedicated to supporting the Drupal project and community. Its main functions are:

- Putting on large conventions around the world
- Maintaining the *Drupal.org* websites and the servers that they run on
- Promoting Drupal as a web platform
- Supporting Drupal education and training
- Providing grants to the Drupal community in support of its mission
- Raising funds for these purposes

Related topics

- [Bagian 1.7, "Concept: Drupal Licensing"](#)
- [Bagian 13.3, "Concept: Security and Regular Updates"](#)
- [Bagian 14.1, "Connecting with the Community"](#)
- [Bagian 14.2, "Getting Support"](#)

Additional resources

- [Drupal project governance](#)

- [*Drupal Code of Conduct*](#)

Attributions

Written by [*Jennifer Hodgdon*](#).

1.7. Concept: Drupal Licensing

Prerequisite knowledge

[*Bagian 1.1, "Concept: Drupal as a Content Management System"*](#)

What is Drupal Licensing?

Drupal and all contributed files hosted on *Drupal.org* are licensed under the GNU General Public License (GPL), version 2 or later. That means you are free to download, reuse, modify, and distribute any files that are part of a project on *Drupal.org* under the terms of GPL version 2 or 3. You can also run the core software in combination with any code with any license that is compatible with version 2 or 3.

Drupal contributors should follow these guidelines:

- All files (PHP, JavaScript, images, Flash, etc.) that are part of a project on *Drupal.org* have to be under GPL version 2 or later.
- All Drupal contributors retain copyright on their code, but agree to release it under the same license as Drupal.
- Drupal modules and themes are a derivative work of Drupal. If you distribute them, you must do so under the terms of GPL version 2 or later.
- All content on *Drupal.org* itself is copyrighted by its original contributors, and is licensed under the Creative Commons Attribution-ShareAlike license 2.0.
- Sample code on *Drupal.org* is also available under GPL version 2 or later.

Related topics

[*Bagian 14.1, "Connecting with the Community"*](#)

Additional resources

- [*Drupal.org page on "Licensing"*](#)
- [*"Drupal 8 LICENSE.txt" page on api.drupal.org*](#)
- [*GNU General Public License, version 2*](#)
- [*GNU General Public License, version 3*](#)

- [*Creative Commons Attribution-ShareAlike license 2.0.*](#)
- [*"Legal Group" on groups.drupal.org*](#)

Attributions

Adapted by [*Diána Lakatos*](#) at [*Pronovix*](#) from [*"Licensing"*](#) copyright 2000-2016 by the individual contributors to the [*Drupal Community Documentation*](#).

Bab 2. Merencanakan Situs Anda

2.1. Concept: Regions in a Theme

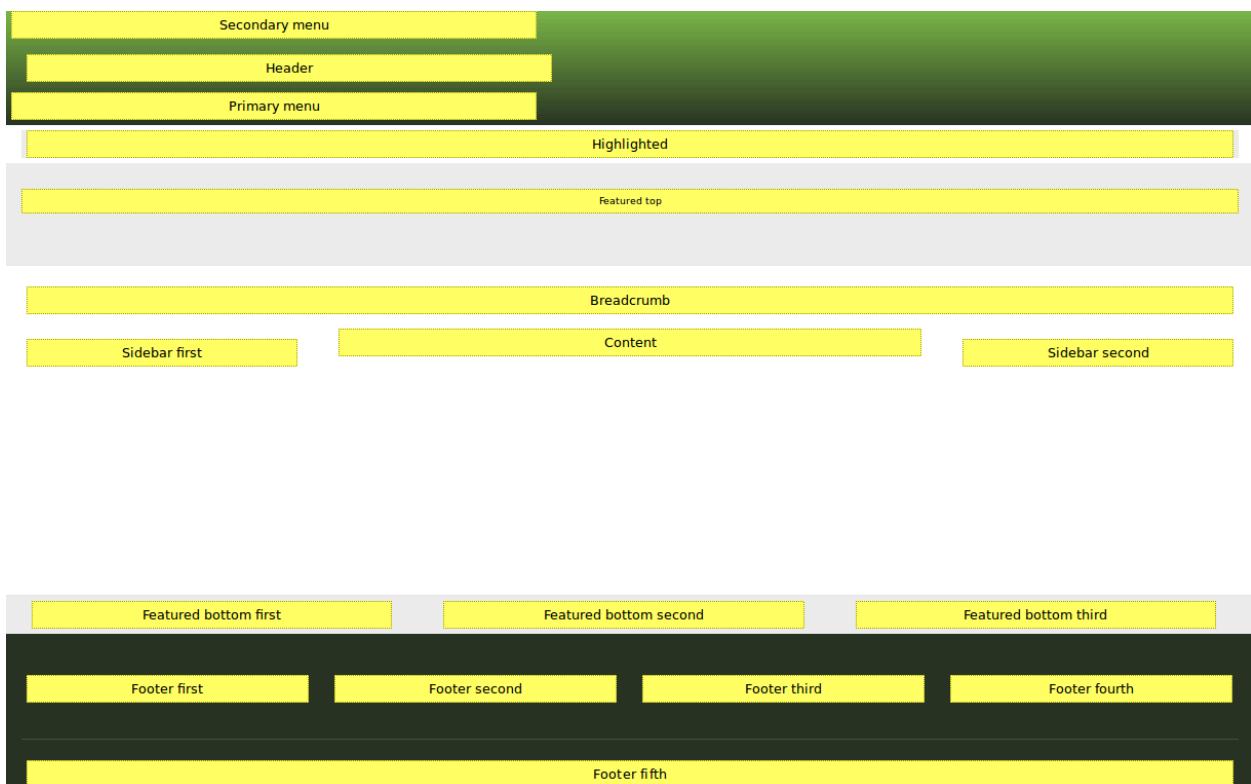
Prerequisite knowledge

[Bagian 1.3, "Concept: Themes"](#)

What is a region?

Besides its primary content, a web page contains other content such as site branding (site name, slogan, and logo), navigation aids (menus, links, and icons), formatted text, and images. Each theme provides a set of named regions, such as *Header*, *Content*, and *Sidebar*, where site builders may choose to place their content.

The available regions depend on the theme design. Only the *Content* region, which contains the primary content, is required; others are optional. The core Bartik theme provides the eighteen regions highlighted in the following image.



Related topics

- [Bagian 8.1, "Concept: Blocks"](#)

- [*Bagian 2.3, "Concept: Content Entities and Fields"*](#)
- [*Bagian 8.3, "Placing a Block in a Region"*](#)

Additional resources

- [*Drupal.org community documentation page "Assigning content to regions"*](#)

Attributions

Written and edited by [*John MacDonald*](#), and [*Michael Lenahan*](#) at [*erdfisch*](#).

2.2. Planning Your Site Layout

Goal

Plan the navigation and layout of the site, for both mobile and desktop browsers.

Prerequisite knowledge

- [*Bagian 1.3, "Concept: Themes"*](#)
- [*Bagian 2.1, "Concept: Regions in a Theme"*](#)
- [*Bagian i.6, "Guiding Scenario"*](#)

Steps

It is a good idea to plan the site layout before you start building the site and writing content; however, your plan may need to be revised either before you start implementing it or after you have some of the site built with draft content in place, based on budgetary concerns or stakeholder feedback.

1. Make a list of the information that your site should present to visitors. In the farmers market scenario, this might include:
 - Location of the market, with directions and a map
 - Hours and days the market is open
 - History of the market
 - List of vendors
 - Details about each vendor
 - Searchable list of recipes
 - Details about each recipe
 - List of the most recently added recipes

2. Decide which information should be on which pages or types of pages on the site:

Information that should be on all pages

Address, hours, and recently-added recipes list

Vendor details pages

Information about each vendor on its own page

Recipe details pages

Details of each recipe on its own page

Home page

Location, map, directions, and hours

About page

History of the market

Vendors list page

List of vendors, with links to vendor detail pages

Recipe list page

Searchable list of recipes, with links to recipe detail pages

3. Decide which information is the most important on each page. Site visitors using mobile phones or other small browsers will often only see the content that is presented first, and they may not scroll down to see all of the information.

4. Decide which of these pages should appear in the main site navigation. For instance, the main navigation might consist of the Home, About, Vendors, and Recipes pages.

5. Make a rough design sketch for each page, showing how it would look when viewed on a small screen such as a phone, as well as on a larger screen such as a desktop browser. Considering that most site visitors will be using smaller browsers, it is a good idea to start with the phone-size layout, to make sure that these visitors will be able to find the information they need without too much scrolling.

In making these page layout plans, you might find that you need to revise your plan for which information should be on which pages. For example, you might decide that the address, hours, and recently-added recipes list would all fit well in the right sidebar area of all pages, when the site is viewed on desktop-sized browsers. On the other hand, you might decide that for mobile browsers, you would instead put the address and hours in a short format at the top of each page, but only display the recent recipe list at the bottom of the home page.

Expand your understanding

[Bagian 2.5, "Planning your Content Structure"](#)

Attributions

Written by [Jennifer Hodgdon](#).

2.3. Concept: Content Entities and Fields

Prerequisite knowledge

- [Bagian 1.5, "Concept: Types of Data"](#)
- [Bagian 1.2, "Concept: Modules"](#)

What is a content entity?

A *content entity* (or more commonly, *entity*) is an item of content data, which can consist of text, HTML markup, images, attached files, and other data that is intended to be displayed to site visitors. Content entities can be defined by the core software or by modules.

Content entities are grouped into *entity types*, which have different purposes and are displayed in very different ways on the site. Most entity types are also divided into *entity sub-types*, which are divisions within an entity type to allow for smaller variations in how the entities are used and displayed. Here is a table of some common content entity types:

Entity type	Entity sub-type	Defining Module	Main uses
Content item	Content type	Node module	Content intended to be the main page area for pages on the site
Example: In the farmers market site example, you might have content types for basic pages, vendor pages, and recipe pages.			
Comment	Comment type	Comment module	Commentary added to content entities (typically to Content item entities)
Example: On a blog site, blog posts might have comments. They are not needed in the farmers market site example.			
User profile	(none)	User module	Data related to a person with a user account (login access) on the site
Example: Every site has at least basic user profiles with user names and email addresses; social networking sites may have more complex user profiles with more information.			
Custom block	Block type	Custom Block module	Text and images in smaller chunks, often displayed in the site

Entity type	Entity sub-type	Defining Module	Main uses
			header, footer, or sidebar
Example: In the farmers market site example, you might put the hours and location in a sidebar block.			
Taxonomy term	Vocabulary	Taxonomy module	Used to classify other types of content
Example: In the farmers market site example, you might classify Recipe content with an Ingredients taxonomy vocabulary, with taxonomy terms like Carrots and Tomatoes. In a blogging site, blog posts might be classified using a Tags vocabulary, and perhaps also a Categories vocabulary.			
File	(none)	File module	An image or attachment file that is tracked and managed by the site, often attached to other types of content
Example: In the farmers market site example, both Recipe and Vendor pages might have image attachments, which would (behind the scenes) be managed as File entities by the site.			
Contact form	Form type	Contact module	A form that lets site visitors contact site owners
Example: A contact form is needed in the farmers market site example.			

What is a field?

Within entity items, the data is stored in individual *fields*, each of which holds one type of data, such as formatted or plain text, images or other files, or dates. Field types can be defined by the core software or by modules.

Fields can be added by an administrator on entity sub-types, so that all entity items of a given entity sub-type have the same collection of fields available. For example, the Vendor content type in the farmers market example might have fields for the vendor name, a logo image, website URL, and description, whereas the *Basic page* content type might only have fields for the title and page body. When you create or edit entity items, you are specifying the values for the fields on the entity item.

Related topics

- [Bagian 2.5, "Planning your Content Structure"](#)
- [Bagian 5.2, "Creating a Content Item"](#)
- [Bagian 6.1, "Adding a Content Type"](#)

- [Bagian 6.5, "Concept: Taxonomy"](#)
- [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)
- [Bagian 8.1, "Concept: Blocks"](#)

Attributions

Written and edited by [Jennifer Hodgdon](#) and [Grant Dunham](#).

2.4. Concept: Modular Content

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 2.2, "Planning Your Site Layout"](#)

What is modular content?

Given that the content of your site is stored in a database, it is desirable to make the content *modular*, meaning that certain pages on your site, rather than being edited as a whole page, are instead generated automatically from other content items. For instance, in the farmers market site scenario, you might create individual content items for recipes. If the recipe content items have a field that keeps track of ingredients, then your site could include a composite page that would list recipes, and allow visitors to search for a recipe that contained some particular ingredient they had bought at the market.

Smaller sections of pages can also be generated as composites. For instance, recipe content items could have a field that keeps track of which vendor submitted the recipe (see [Bagian 6.4, "Concept: Reference Fields"](#)), with the vendor details edited in separate vendor content items. This would allow you to do the following on your site:

- On each Recipe page, there could be an area that displays some information about the vendor that submitted the recipe, such as their name and market stall number.
- Each vendor page could have a section that lists the recipes they have submitted.

The key idea is that each piece of information is only edited in one place. When vendor information is updated, all recipe pages that display that vendor information are automatically updated; when a recipe is submitted by a vendor, it is automatically displayed on the vendor page. The core Views module is the usual way to use modular content to create composite pages and page sections; see [Bagian 9.1, "Concept: Uses of Views"](#) for more information. Also, *view modes* are useful for defining different ways to display each content item; see [Bagian 6.10, "Concept: View Modes and Formatters"](#) for more information.

Related topics

- [Bagian 2.5, "Planning your Content Structure"](#)

- [*Bagian 6.1, "Adding a Content Type"*](#)
- [*Bagian 6.3, "Adding Basic Fields to a Content Type"*](#)
- [*Bagian 6.4, "Concept: Reference Fields"*](#)
- [*Bagian 6.10, "Concept: View Modes and Formatters"*](#)
- [*Bagian 9.1, "Concept: Uses of Views"*](#)

Attributions

Written by [*Jennifer Hodgdon*](#).

2.5. Planning your Content Structure

Goal

Make a plan for the content structure of the site (which type and subtype of entity to use for which content), and which pages will contain listings of content.

Prerequisite knowledge

- [*Bagian 2.3, "Concept: Content Entities and Fields"*](#)
- [*Bagian 2.4, "Concept: Modular Content"*](#)
- [*Bagian i.6, "Guiding Scenario"*](#)

Steps

1. Brainstorm about what content your site needs to contain, which could include content that visitors would be looking for, as well as content that you want to show to visitors. The result could be the description in [*Bagian i.6, "Guiding Scenario"*](#).
2. For each identified piece of content, decide which content entity type would be the best fit. In doing this, you'll need to consider where and how the content will be used and edited on the site. For example, in the farmers market site scenario, you might want to display the hours and location of the farmers market on the sidebar of every page. For that content, a single custom block makes sense. As another example, you might decide that pages displaying information about each vendor should be content items managed by the core Node module, because you want vendors to be able to edit their own listings. The core Node module permission system lets you do this easily.

These decisions do not necessarily always have only one right answer; for instance, you could decide that vendor pages should be user profiles instead of content items, but if you did that the content would be tied to a specific user account, and it would not be as easy to later change the ownership of a vendor page to a different user account.

3. Within each content entity type you identified, decide what division into entity sub-types would make sense. For example, in the farmers market site example, you would proba-

bly decide that under the Content item entity type, there should be one content type for basic pages (Home and About), one for vendor pages, and one for recipe pages.

4. For each entity sub-type you decided on, decide what fields are needed. For instance, the Vendor content type might need fields for the vendor name, web page URL, image, and description.
5. Decide on what entity listings are needed, which could be entire pages or smaller areas on the page. For each listing, you'll need to determine what entity items should be listed. Then you'll need to decide in what order and with what filtering options they should be displayed; for example, you might want to give the site visitor the option to search by keyword, to filter the list down to a subset, or to sort the list. You'll also need to decide what information from the entity items should be shown, which might result in adding to the list of fields you determined in the previous step. The farmers market site, for example, needs to have a Recipes listing page that lists content items of type Recipe, with the ability to filter by ingredients, so that means that the Recipe content type needs an Ingredients field.
6. For each identified field on each entity subtype, identify what type of data it should contain (such as plain text, formatted text, a date, an image file, etc.), and how many values should be allowed. Most fields are single-valued, but for example, a Recipe should allow for multiple values in its Ingredients field.
7. Consider which fields would be best as references to taxonomy term entities: fields whose values should be chosen from a list of allowed values. Allowed values that are expected to change and grow over time, are good candidates. An example is the Ingredients field for the Recipe content type.
8. Consider which fields should reference other content entities. An example is that since vendors will be submitting recipes, a field will be needed on the Recipe content type that references the Vendor content item for the vendor who submitted the recipe.

Here's an example of the resulting content structure for the farmers market scenario example site:

Entity type	Entity sub-type	Examples	Fields
Content item	Basic page	Home page, about page	Title, page body
Content item	Vendor	A page for each vendor at the market	Vendor name, page body, image, URL
Content item	Recipe	A page for each submitted recipe	Recipe name, page body, image, reference to Vendor who submitted it, Ingredients taxonomy
Custom block	(generic)	Copyright notice for footer, Hours and location for sidebar	No special fields
Taxonomy term	Ingredients	Carrots, tomatoes, and other recipe ingredients	No special fields

Entity type	Entity sub-type	Examples	Fields
Contact form	(generic)	Generic contact form	Name, email, subject, message
User profile	(none)	Will not be displayed on site	No special fields

And here are the listings the site needs:

Page or page area	Entity type and sub-type	Filter/sort/pagination	Fields displayed
Vendors page	Vendor content items	All vendors, alphabetical, paged	Image, vendor name, trimmed body
Recipes page	Recipe content items	Filter by ingredients, alphabetical, paged	Image, recipe name
Recent recipes sidebar	Recipe content items	List 5 most recent	Image, recipe name

Expand your understanding

- [*Bagian 6.1, "Adding a Content Type"*](#)
- [*Bagian 6.3, "Adding Basic Fields to a Content Type"*](#)
- [*Bagian 6.6, "Setting Up a Taxonomy"*](#)

Related concepts

[*Bagian 6.5, "Concept: Taxonomy"*](#)

Attributions

Written and edited by [*Jennifer Hodgdon*](#) and [*Grant Dunham*](#).

2.6. Concept: Editorial Workflow

Prerequisite knowledge

[*Bagian 1.1, "Concept: Drupal as a Content Management System"*](#)

What is an Editorial Workflow?

An Editorial Workflow is the process organizations follow to create, review, edit and publish content.

Depending on the size and processes of the organization, multiple people in different roles can be part of the process. For example, content creators collect information and write content, editors review, edit, ask for changes, and publish the content once it's considered ready to be shared with the audience.

In your site, content types can have either a Published or an Unpublished flag to track their workflow states, that you can set for each content item when saving:

- Published content items are visible to all visitors of the site.
- Unpublished content items are not visible to the visitors of the site, but can be seen by authenticated users who have the permissions to view unpublished versions of specific content items.

Most content entity types support revision tracking, if it is enabled by an administrator for the entity sub-type. If you have revision tracking enabled, the software will store old versions of the content each time it's revised, and add notes about what is changed. This allows you to look at the history and revert to older versions.

Using the Unpublished flag and Revisions, you can build an editorial workflow that lets content creators and editors work on the live site without displaying the content to visitors. When the content is ready to be published, all they need to do is save it as Published.

Related topics

- [*Bagian 5.2, "Creating a Content Item"*](#)
- [*Bagian 5.3, "Editing a Content Item"*](#)

Additional resources

Talking about online content, organizations would also want to consider post-publication phases of the editorial workflow: content teams might need to update already published content. Some organizations might need a more complex workflow with the options to configure which content type goes through the workflow process, set up user roles and permissions, set different workflow states (for example draft, needs review, published).

For more complex workflows, you can add install or more modules from the [contributed Workbench Suite module](#) to your site.

Attributions

Written and edited by [Diána Lakatos](#) at [Pronovix](#), and [Grant Dunham](#).

Bab 3. Instalasi

3.1. Concept: Server Requirements

What are the requirements for running the core software?

Disk space

A minimum installation requires 15 MB. 60 MB is needed for a website with many contributed modules and themes installed. Keep in mind you need much more for the database, files uploaded by the users, media, backups, and other files.

Web server

Apache (Recommended)

Apache is the most commonly used web server. The core software will work on Apache 2.x hosted on UNIX/Linux, OS X, or Windows that have the Apache mod_rewrite module installed and enabled. The Apache VirtualHost configuration must contain the directive *AllowOverride All* to allow the *.htaccess* file to be used.

Nginx

Nginx is a commonly used web server that focuses on high concurrency, performance and low memory usage. The core software will work on Nginx 1.1 or greater hosted on UNIX/Linux, OS X, or Windows. The *ngx_http_rewrite_module* must be installed and enabled.

Hiawatha

Hiawatha is a web server with a strong focus on security. It also aims at being easy to use and lightweight. Research performed by independent researchers proves that Hiawatha offers a performance comparable to other web servers under normal conditions, but much better while under attack. URL Toolkit must be installed and configured to rewrite URLs.

Microsoft IIS

Microsoft IIS is a web server and set of feature extension modules for use with Microsoft Windows. The core software will work with IIS 5, IIS 6, or IIS 7 if PHP is configured correctly. Because clean URLs are required, you may need to use a third party product. For IIS 7, you can use the Microsoft URL Rewrite module or a third party solution.

Database

Use one of the following databases:

- MySQL - 5.5.3 (MariaDB 5.5.20, Percona 5.5.8) or higher with an InnoDB-compatible primary storage engine

- PostgreSQL - 9.1.2 or higher
- SQLite - 3.4.2 or higher

PHP

PHP 5.5.9 or higher.

Additional resources

[Drupal.org community documentation page "System requirements"](#)

Attributions

Adapted by [Karl Kedrovsky](#) and [Brian Emery](#) from ["System requirements"](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

3.2. Concept: Additional Tools

What tools are available for site builders?

There are several additional tools available that help you create sites faster, more accurately and with less effort.

Drush

See below for more about Drush.

Git

[Git](#) is a version control system you can use to track changes made to files within your entire site, or files relating only to the module or theme that you are using. Git makes it possible to try out changes, and if necessary, revert to a previous version. It can be used as well to track changes in the configuration files and synchronize the configuration of different sites. Git has become the standard for web development projects.

Composer

See below for more about Composer.

Devel

The [contributed Devel module](#) helps with development tasks such as debugging and inspecting code, analyzing database queries, and generating dummy content.

What is special about Drush?

[Drush](#) is a command-line tool for building and maintaining sites, providing an alternative to using the administrative interface for many operations. Many site builders and maintainers prefer to use Drush whenever possible, and consider the time spent installing and learning to use Drush to be a good investment. Reasons to consider using Drush:

- Drush provides a large number of commands for tasks such as installing the base software, modules, and themes; updating software; performing database operations, including copying and querying; resetting passwords; and clearing the cache. Some contributed modules and themes also define Drush commands.
- Doing administrative tasks using Drush is typically faster and less tedious than performing the same tasks through the administrative interface.
- Since Drush is a command-line interface, you can write scripts that combine Drush commands and other commands on the server, to automate more complicated tasks.
- Drush provides additional functionality not available via the administrative interface; for example, running database queries.

If you want to use Drush, you will need to have command-line terminal access to the server where your website will be hosted, and you will need to have a compatible version of the Drush tool installed on that server. Check the [Drush website](#) for installation instructions and version compatibility. There is also documentation on the available Drush commands on that site.

What is Composer used for?

[Composer](#) is a tool for managing PHP dependencies, where the developer specifies what version of each external library is needed, and Composer manages the process of downloading and installing the libraries.

The core software is a primary user of Composer, because it makes use of several externally-developed software libraries, which must be downloaded and installed in order for the core software to work. When you install the core software, you either need to download an archive that contains compatible versions of the external libraries, or you need to run Composer to download the external libraries after the initial download.

Some contributed modules also make use of externally-developed software libraries; for example, a Facebook integration module might require Facebook's integration library to be installed for the module to work, and a geographical module might make use of a standard library of geographical functions. To install a module with external dependencies, you will need to run Composer.

What tools are available for module and theme developers?

In addition to the site builder tools mentioned above, the following tools are useful for module and theme developers.

Drupal Console

[Drupal Console](#) is a command-line tool that generates boilerplate code and interacts with a Drupal site. It can generate, for example, block or form code, install modules and themes, and create dummy content. Drupal Console makes use of the Symfony Console.

Coder

[Coder](#) is a command-line tool that checks if your modules and themes comply with coding standards and other best practices. It can also fix coding standard violations.

Firebug

[Firebug](#) is a Firefox plugin that allows viewing, editing, debugging, and monitoring CSS, HTML, and JavaScript in a separate browser window. The [contributed Drupal for Firebug module](#) displays Drupal debugging and SQL query information in the Firebug window.

Related topics

[Bagian 11.9, "Using Composer and Git to Download Files"](#)

Additional resources

- [Drupal.org community documentation page "Development tools"](#)
- [Drupal.org community documentation page "Using Composer with Drupal"](#)

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

3.3. Preparing to Install

Goal

Choose an appropriate way to install the core software, and handle any required prerequisites.

Prerequisite knowledge

- [Bagian 3.1, "Concept: Server Requirements"](#)
- [Bagian 3.2, "Concept: Additional Tools"](#)

Site prerequisites

If you want to use Drush to install the core software, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

1. Choose a method for installing the core software.

Try a free online demo

Use an online provider to get a demo installation of the core software in 20 minutes or less. See the [Drupal.org page "Try Drupal!"](#).

Use a one-click installer from your hosting provider

If you choose to install the core software at your hosting provider, your hosting provider may have specific documentation and/or a one-click install that you can use. See [Drupal.org's list of hosting providers that support Drupal](#).

Use Drush

You will need to have Drush installed and command-line access to the site. See [Bagian 3.2, "Concept: Additional Tools"](#).

Use the web installer

You will need space on a local or hosted web server to install the core software manually.

Use a pre-configured environment

Use a pre-configured environment or virtual machine that contains Drupal and all the required supporting software to install Drupal locally. See the section for your operating system under [Drupal.org's Local server setup guide](#) for possible options.

2. If you are installing using Drush or the web installer, set up a URL and hosting for your site, either at an external hosting provider or on your own computer. Verify that the hosting is working by putting a simple HTML file in the web root directory of the hosting, and visiting the URL for your site.
3. If you are installing using Drush or the web installer, create a database, along with a database user account with full access.
4. If you are installing using Drush, type the following at the Drush command line, where *example* is the name of the directory that the core software will be downloaded to, and *DB_NAME*, *DB_USER* and *DB_PASS* are your database's credentials:

```
drush dl drupal --drupal-project-rename=example  
cd example  
drush site-install standard --db-url='mysql://DB_USER:DB_PASS@localhost/DB_NAME'
```

5. If you are installing using the web installer, upload the core software files manually. To do this using Git and/or Composer, see [Bagian 11.9, "Using Composer and Git to Download Files"](#); to do this by downloading a zipped archive:
 - a. Go to <https://www.drupal.org> and click *Download & Extend* on the top-level menu.

Download & Extend

Download

[Download Drupal 8.1.10](#)

[Try a hosted Drupal demo](#)

See [Drupal's project page](#) for more information, older versions, and project development. Explore [hosting options](#) for your site. Browse [documentation](#) for more help and information.

- b. Click the *Download Drupal* button (of the appropriate version) to be taken to the Downloads page.

Downloads

Recommended releases

These are stable, well-tested versions that are actively supported.

Drupal core 8.1.4

Released: Jul 06 2016

The next patch release of Drupal 8 is ready for new development and use on production sites.

Drupal core 7.44

Released: Jun 15 2016

If you need stability and features from the widest variety of contributed modules and themes, this is the version for you.

- c. Click the version you wish to download under *Recommended releases*.

Releases

drupal 8.1.3

[Download drupal-8.1.3.tar.gz tar.gz](#)

11.97 MB

[Download drupal-8.1.3.zip zip](#)

19.55 MB

- d. Download the tar.gz or zip file to your local machine.
- e. Upload the downloaded file to your hosting account. Log in to the control panel and navigate to the *HTML* directory. Save the file there.

- f. Uncompress the tar.gz or zip file, which will create a new directory. If you do not have terminal access, or your hosting server is not running Linux, your hosting control panel's file manager should provide a way to extract the files. If you have terminal access to your hosting server (running Linux), you can use a command like:

```
tar -xzf drupal-8.3.2.tar.gz
```

- g. Delete the compressed file from the server, unless your unpacking method already deleted it.
- h. Rename the directory or reconfigure your web hosting, so that the directory name matches the directory your web hosting is configured to use for your site.

Expand your understanding

Run the installer if you chose web installer to install the core software. See [Bagian 3.4, "Running the Installer"](#).

Additional resources

[Drush](#)

Attributions

Written and edited by [Drew Gorton](#), [Michael Lenahan](#) at [erdfisch](#), [Jennifer Hodgdon](#), and [Joy Alphonso](#) at [Red Crackle](#).

3.4. Running the Installer

Goal

Install the core software and create the admin account by running the included installer.

Prerequisite knowledge

[Bagian 3.1, "Concept: Server Requirements"](#)

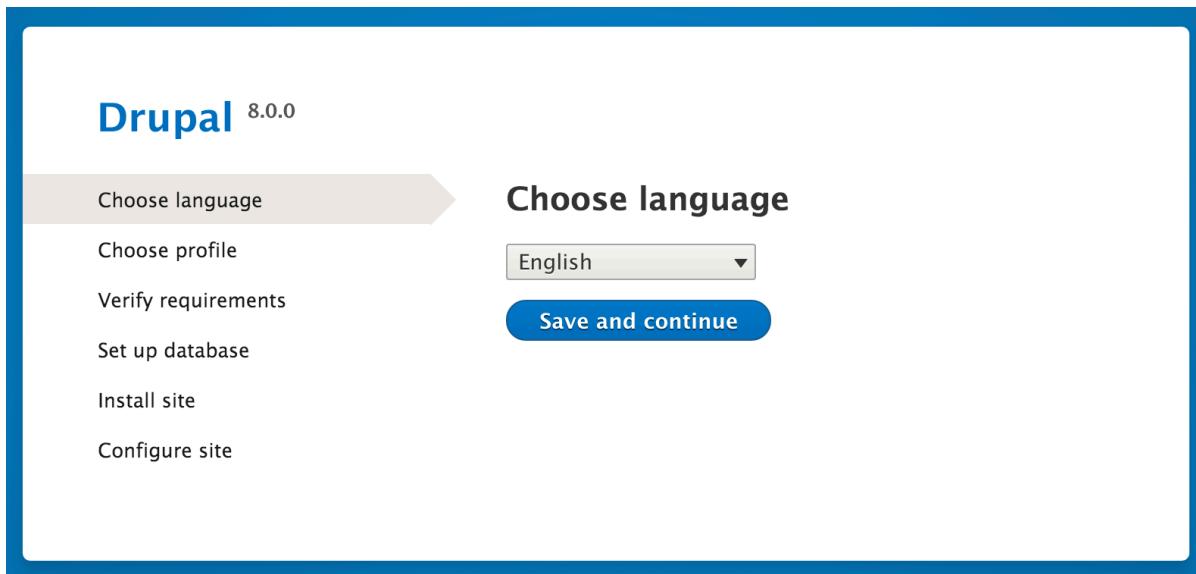
Site prerequisites

[Bagian 3.3, "Preparing to Install"](#)

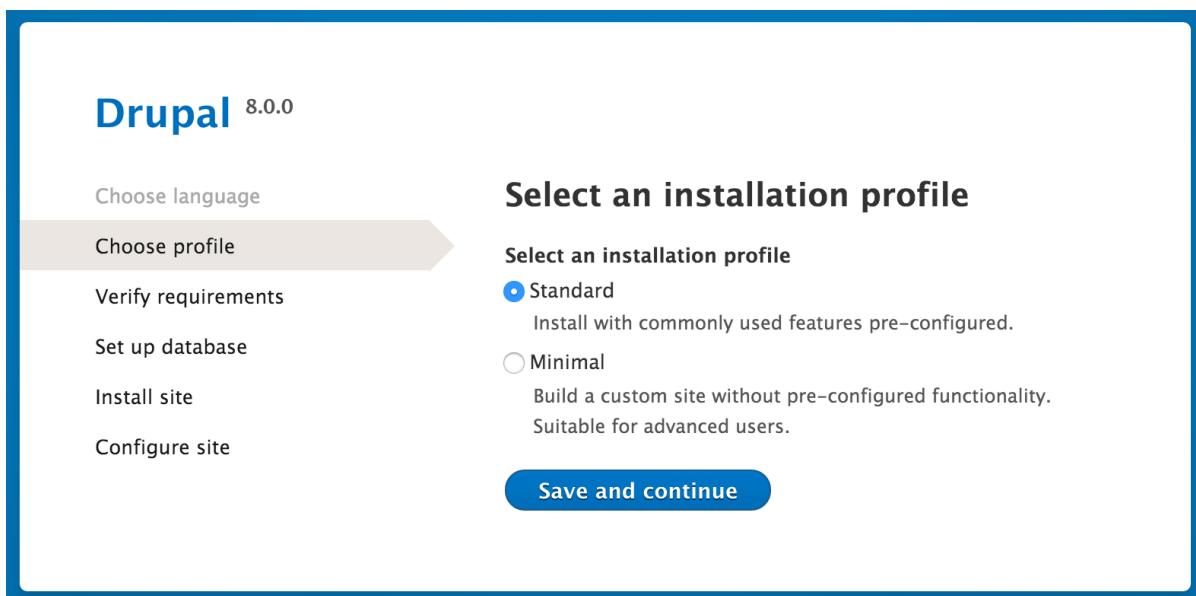
Steps

1. If you are using a 1-click install from a hosting provider or demo site, you will most likely see some or all of the following screens as part of the installation process. If you uploaded the core files manually, to start the installer, open a browser and visit the URL that you set up for your hosting.

2. Select a language on the first page of the installer; for example, English. You could optionally choose from any of the other listed languages. The language files for the chosen language will be downloaded and installed so that the rest of the installation process can be finished in the chosen language. After choosing a language, click *Save and continue*.



3. Select an installation profile. Installation profiles provide site features and functions for a specific type of site as a single download containing the core software, contributed modules, themes, and pre-defined configuration. Core contains two installation profiles. Select the core Standard installation profile. Click *Save and continue*.



4. The next step in the installer will verify that your system meets the minimum requirements. If it does not, you'll be presented with an outline of what needs to be corrected in order to proceed. If it does, the installer will automatically advance to the next step.

5. Provide details of the database you created in the *Bagian 3.3, "Preparing to Install"* chapter. Then click *Save and continue*.

Field name	Explanation	Value
Database name	The custom name given to the database	drupal8
Database username	Username created	databaseUsername
Database password	Password chosen	

Drupal 8.0.1

Choose language

Choose profile

Verify requirements

Set up database

Install site

Configure site

Database configuration

Database type *

MySQL, MariaDB, Percona Server, or equivalent

Database name *

drupal9 *

Database username *

databaseUsername *

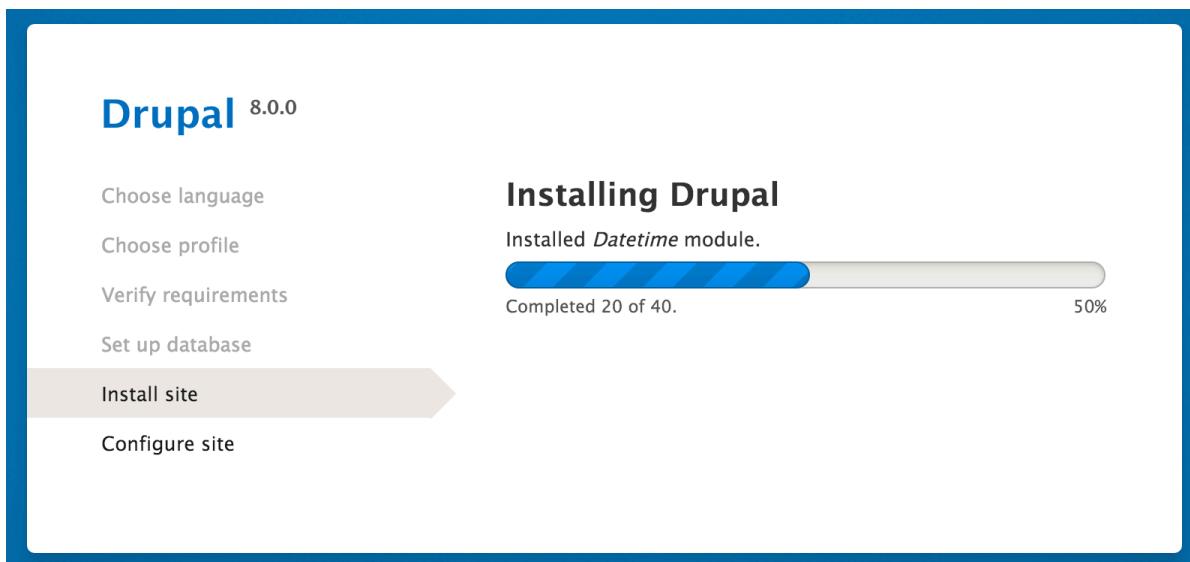
Database password

***** *

► ADVANCED OPTIONS

Save and continue

6. The next step will display a progress bar under the heading *Installing Drupal*. After the installer has completed, it will automatically advance to the next step.



7. The final step is to configure some basic information about your new site. Note that the user account you create in this step is the site's admin account. See [Bagian 7.2, "Concept: The User 1 Account"](#) for important information about this unique account. You can safely name this account "admin", and make sure to choose a secure and unique password.

Fill in the form with the following information:

Field name	Explanation	Value
Site name	The name chosen for the site	Anytown Farmers Market
Site email address	The email associated with the site	info@example.com
Username	The designated user's credentials	admin
Password	The password chosen	
Confirm password	Repeat the password	
Email address	The user's email	admin@example.com

The remaining fields can likely be left at their default values.

Drupal 8.0.1

[Choose language](#)
[Choose profile](#)
[Verify requirements](#)
[Set up database](#)
[Install site](#)
Configure site

Configure site

SITE INFORMATION

Site name *
 [i]

Site email address *

Automated emails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these emails from being flagged as spam.

SITE MAINTENANCE ACCOUNT

Username *

Several special characters are allowed, including space, period (.), hyphen (-), apostrophe ('), underscore (_), and the @ sign.

Password *
 [i]
 Password strength: Strong

Confirm password *
 [i]

Passwords match: yes

To make your password stronger:

- Add uppercase letters

Email address *

REGIONAL SETTINGS

Default country
 ▼

Select the default country for the site.

Default time zone
 ▼

By default, dates in this site will be displayed in the chosen time zone.

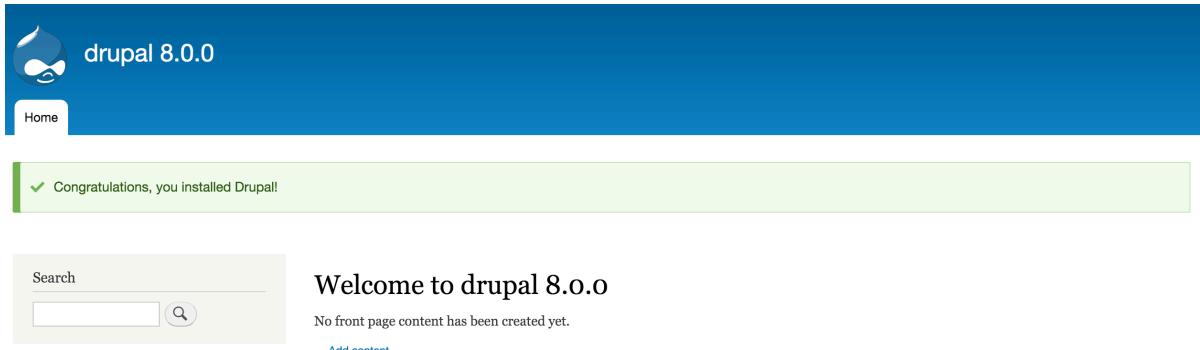
UPDATE NOTIFICATIONS

Update notifications

Check for updates automatically
 Receive email notifications

The system will notify you when updates and important security releases are available for installed components. Anonymous information about your site is sent to [Drupal.org](#).

8. Click *Save and continue*.
9. You will be redirected to the front page of your new site and you should see the message *Congratulations, you installed Drupal!* displayed at the top of the page.



Expand your understanding

Check the Status Report to see if there are any problems with the installation. See [*Bagian 12.5, "Concept: Status Report"*](#).

Related concepts

- [*Bagian 11.7, "Concept: Development Sites"*](#)
- [*Bagian 3.2, "Concept: Additional Tools"*](#)

Additional resources

[*Drupal.org community documentation page "Create A Database"*](#)

Attributions

Written and edited by [*Joe Shindelar*](#) at [*Drupalize.Me*](#), and [*Jojo Alphonso*](#) at [*Red Crackle*](#).

Bab 4. Konfigurasi Dasar Situs

4.1. Concept: Administrative Overview

Prerequisite knowledge

- [Bagian 1.3, "Concept: Themes"](#)
- [Bagian 1.2, "Concept: Modules"](#)

What is the administrative menu?

The toolbar provided by the core Toolbar module displays the *Manage* administrative menu at the top or left side of the site, for users with permission to see it. This menu provides access to all of the administrative areas of the site. The menu entries will vary depending on which modules are active on your site and the permissions of the person viewing the menu; if you install using the core Standard installation profile and have full administrative permissions, the top-level entries are as follows:



Content

Lists and manages existing content, and allows creation of new content.

Structure

Contains a list of links for managing structural elements of the site, such as blocks, content types, menus, and taxonomy.

Appearance

Manages themes and appearance-related settings.

Extend

Manages the installation and uninstallation of modules.

Configuration

Contains links to settings pages for various site features.

People

Manages users, roles, and permissions.

Reports

Contains links to logs, update information, search information, and other information about the site's status.

Help

Lists help topics for installed modules that provide them.

The arrow button on the far right side of the second line of the toolbar (or far left side, if the site is being viewed using a right-to-left-reading language like Arabic) can be used to switch the menu from appearing horizontally at the top of the page, to a vertical format on the left side (or right side, in right-to-left languages). When viewed vertically, the menu becomes an interactive tree.

The screenshot shows the top navigation bar of a Drupal site. The 'Manage' tab is highlighted in blue, while the 'Shortcuts' tab is also present. Below the toolbar, there is a vertical list of administrative links:

- Content
- Structure
- Appearance
- Extend
- Configuration
- People
- Reports

Below the 'Reports' link, there are several status messages and links:

- Available updates
- Recent log messages
- Available translation updates
- Field list
- Status report

A small note at the bottom left says "Top 'Access denied'".

This guide has a standard way to describe navigation to administrative pages using the administrative toolbar. See [Bagian i.5, "Conventions of the Guide"](#) for more information.

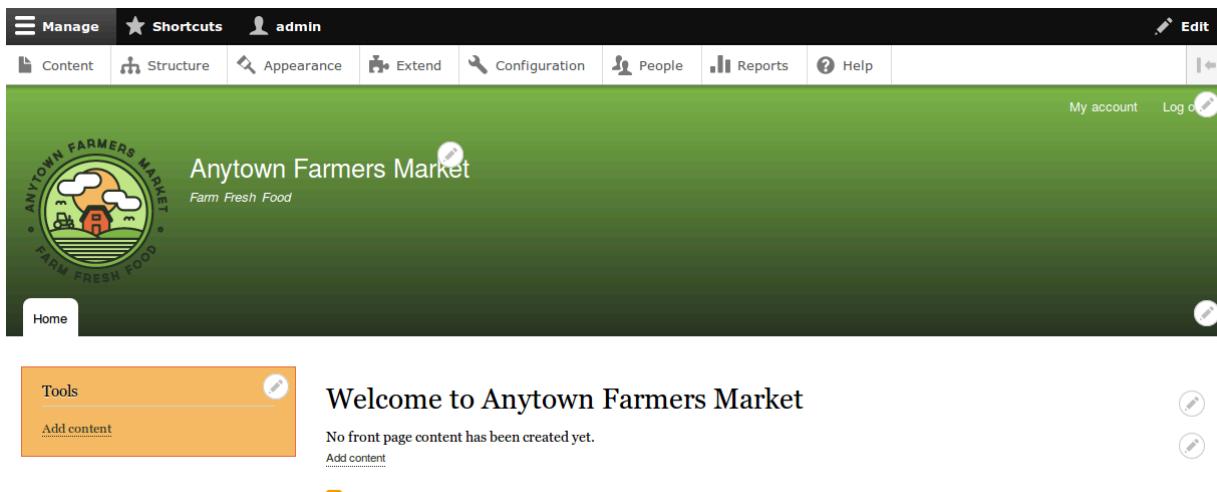
What are contextual links?

Some administrative and editing functionality on the site can be accessed through the *contextual links* displayed by the core Contextual Links module. Contextual links take you to so-

me of the same pages that you can access through the administrative menu, but instead of having to navigate through the menu hierarchy, these links are provided near where the related content is displayed on your site.

Contextual links have to be activated to be visible. If your site's theme uses the default styling for contextual links, a pencil icon is used to indicate that contextual links are present and activated, and if you click the icon, you will see the contextual links. There are two ways to activate the pencil icons that provide access to the contextual links:

- If you are using a mouse in a browser, the icon will temporarily appear when you hover over an area that has related contextual links.
- You can click the master pencil icon (or its *Edit* link) at the right end of the top bar in the toolbar, which will activate all of the contextual links on the current page. This icon is only visible on pages with contextual links.



Attributions

Written by [Scott Wilkinson](#) and [Jennifer Hodgdon](#).

4.2. Editing Basic Site Information

Goal

Change basic site information such as *Site name*, *Slogan*, *Default time zone*.

Prerequisite knowledge

[Bagian 4.1, "Concept: Administrative Overview"](#)

Steps

Configuring the basic site information

1. In the *Manage* administrative menu, navigate to *Configuration > System > Basic site settings* (`admin/config/system/site-information`) to change the *Site name*, *Slogan*, administrative *Email address*, or the *Default front page* path.
2. Fill in the available fields as appropriate for your site.

Field name	Explanation	Example value
Site name	Used to identify the site and displayed in browsers	Anytown Farmers Market
Slogan	Usually displayed sitewide	Farm Fresh Food
Email address	Used as <i>From</i> address in automated email messages (registrations, password resets, etc)	info@example.com

▼ SITE DETAILS

Site name *
Anytown Farmers Market

Slogan
Farm Fresh Food
How this is used depends on your site's theme.

Email address *
info@example.com
The *From* address in automated emails sent during registration and new password requests, and other notifications. (Use an address ending in your site's domain to help prevent this email being flagged as spam.)

3. After editing the fields, click *Save configuration* to see the changes applied to the site.

Configuring default Regional settings

1. In the *Manage* administrative menu, navigate to *Configuration > Regional and language > Regional settings* (`admin/config/regional/settings`).
2. Select the appropriate country from the *Default country* select list in the *Locale* section, and set which day will be displayed as *First day of week* on your site.
3. In the *Time zones* section, select the sitewide *Default time zone* from the list and configure user-specific time zones.

LOCALE

Default country
United States

First day of week
Sunday

TIME ZONES

Default time zone
America/Los Angeles

Users may set their own time zone

Remind users at login if their time zone is not set
Only applied if users may set their own time zone.

Time zone for new users

Default time zone

Empty time zone

Users may set their own time zone at registration
Only applied if users may set their own time zone.

Save configuration

4. After editing the fields, click *Save configuration* to see the changes applied to the site.

Additional resources

[Drupal.org community documentation page "Getting started with Drupal 8 administration"](#)

Attributions

Written and edited by [Sree Veturi](#), [Michael Lenahan](#) at [erdfisch](#), and [Antje Lorch](#).

4.3. Installing a Module

Goal

Install a core module, or a contributed module whose files have already been uploaded to the site, through the administrative interface or using Drush.

Prerequisite knowledge

[Bagian 1.2, "Concept: Modules"](#)

Site prerequisites

If you want to use Drush to install modules, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

You can use the administrative interface or Drush to install modules.

Using the administrative interface

1. In the *Manage* administrative menu, navigate to *Extend (admin/modules)*. The *Extend* page appears showing all the available modules in your site.
2. Check the boxes for the module or modules you want to install. For example, check the box for the core Activity Tracker module.

CORE	
<input type="checkbox"/>	Actions
	▶ Perform tasks on specific events triggered within the system.
<input checked="" type="checkbox"/>	Activity Tracker
	▶ Enables tracking of recent content for users.
<input type="checkbox"/>	Aggregator
	▶ Aggregates syndicated content (RSS, RDF, and Atom feeds) from external sources.
<input checked="" type="checkbox"/>	Automated Cron
	▶ Provides an automated way to run cron jobs, by executing them at the end of a server response.
<input type="checkbox"/>	Ban
	▶ Enables banning of IP addresses.

3. Click *Install*. The checked modules will be installed.

Using Drush

1. In the *Manage* administrative menu, navigate to *Extend (admin/modules)*. The *Extend* page appears showing all the available modules in your site.
2. Find the machine name of the module you want to install, by expanding the information area for the module. For instance, the core Activity Tracker module's machine name is *tracker*.
3. Run the following Drush command to install the module:

```
drush en tracker
```

Expand your understanding

If you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Additional resources

[Drush](#)

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

4.4. Uninstalling Unused Modules

Goal

Uninstall the core Search and History modules, as well as the core Activity Tracker module if you installed it in [Bagian 4.3, "Installing a Module"](#), to reduce overhead.

Prerequisite knowledge

[Bagian 1.2, "Concept: Modules"](#)

Site prerequisites

- You must have at least one unused module on your site that you want to uninstall, such as the core Search module.
- If you want to use Drush to uninstall modules, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

You can use the administrative interface or Drush to uninstall modules.

Using the administrative interface

1. In the *Manage* administrative menu, navigate to *Extend > Uninstall* ([admin/modules/uninstall](#)) where you will find the list of modules that are ready to be uninstalled.
2. Check the boxes for the modules you are uninstalling (*Search*, *History*, and *Activity Tracker*). Click *Uninstall* at the bottom of the page.

UNINSTALL	NAME	DESCRIPTION
<input checked="" type="checkbox"/>	Activity Tracker	Enables tracking of recent content for users.
<input type="checkbox"/>	Automated Cron	Provides an automated way to run cron jobs, by executing them at the end of a server response.
<input type="checkbox"/>	Block	Controls the visual building blocks a page is constructed with. Blocks are boxes of content rendered into an area, or region, of a web page. The following reason prevents Block from being uninstalled: <ul style="list-style-type: none">• Required by: Custom Block
<input type="checkbox"/>	Breakpoint	Manage breakpoints and breakpoint groups for responsive designs. The following reason prevents Breakpoint from being uninstalled: <ul style="list-style-type: none">• Required by: Toolbar

Catatan

You cannot uninstall a module if it is required by some other module(s) and/or functionality. For example, the core File module is required by the core Text Edi-

tor, CKEditor, and Image modules. It can't be uninstalled unless you uninstall its dependent module(s) and functionality first. A module that cannot be uninstalled yet will have a disabled checkbox, restricting you from uninstalling it.

3. Step 2 will prompt you to confirm the module uninstall request. Click *Uninstall*.

The screenshot shows a confirmation dialog titled "Confirm uninstall". The URL in the header is "Home » Administration » Extend » Uninstall". The main message says: "The following modules will be completely uninstalled from your site, and *all data from these modules will be lost!*" followed by a list of modules: Activity Tracker, History, and Search. Below this, under "CONFIGURATION DELETIONS", it says "The listed configuration will be deleted." and lists "Block" (Search) and "Search page" (Content, Users). At the bottom, the question "Would you like to continue with uninstalling the above?" is followed by two buttons: "Uninstall" (blue) and "Cancel".

Using Drush

1. In the *Manage* administrative menu, navigate to *Extend (admin/modules)*. The *Extend* page appears showing all the available modules in your site.
2. Find the machine name of the module you want to uninstall, by expanding the information area for the module. For instance, the core Activity Tracker module's machine name is *tracker*.
3. Run the following Drush command to uninstall the module:

```
drush pm-uninstall tracker
```

Expand your understanding

- [*Bagian 3.2, "Concept: Additional Tools"*](#)

- [Bagian 12.2, "Clearing the Cache"](#)
- You can also uninstall the core Comment module by following these steps, but only after comment fields have been removed, which is a side effect of [Bagian 6.2, "Deleting a Content Type"](#).

Attributions

Written and edited by [Surendra Mohan](#), and [Jojo Alphonso](#) at [Red Crackle](#).

4.5. Configuring User Account Settings

Goal

Turn off the ability for people to register user accounts on the site. Also, review and/or edit the email messages generated by the site for events related to user accounts.

Prerequisite knowledge

[Bagian 4.1, "Concept: Administrative Overview"](#)

Steps

1. In the *Manage administrative* menu, navigate to *Configuration > People > Account settings (admin/config/people/accounts)*.
2. Under *Registration and cancellation*, select *Administrators only* as the people with permissions to register user accounts. You can check *Require email verification when a visitor creates an account* in case you want to change the settings for account registration later on.

▼ REGISTRATION AND CANCELLATION

Who can register accounts?

Administrators only
 Visitors
 Visitors, but administrator approval is required

Require email verification when a visitor creates an account
New users will be required to validate their email address prior to logging into the site, and will be assigned a system-generated password. With this setting disabled, users will be logged in immediately upon registering, and may select their own passwords during registration.

Enable password strength indicator

When cancelling a user account

Disable the account and keep its content.
 Disable the account and unpublish its content.
 Delete the account and make its content belong to the *Anonymous* user.

Users with the *Select method for cancelling account* or *Administer users permissions* can override this default method.

3. Optionally, change the default email address from which user account notifications from the farmers market website will be sent. This will help you maintain a separate email address from the one used for the website in general. For example, this email address for user account notifications will be useful for a staff member(s) communicating with vendors.

Notification email address

The email address to be used as the 'from' address for all account notifications listed below. If '*Visitors, but administrator approval is required*' is selected above, a notification email will also be sent to this address for any new registrations. Leave empty to use the default system email address (info@example.com).

4. Optionally, edit the email templates in the *Emails* section to customize automated emails. There are seven email templates available with the core. They are meant for different user-specific occasions. All of them can be personalized and three can be disabled via checkboxes: activation, blocking, and cancellation.

You can send out your own text (for example, welcoming the new vendors for whom accounts were just created) by editing the the *Welcome (new user created by administrator)* template.

Emails

Welcome (new user created by administrator)	WELCOME (NEW USER CREATED BY ADMINISTRATOR) Edit the welcome email messages sent to new member accounts created by an administrator. Available variables are: [site:name], [site:url], [user:display-name], [user:account-name], [user:mail], [site:login-url], [site:url-brief], [user:edit-url], [user:one-time-login-url], [user:cancel-url]. Subject An administrator created an account for you at [site:name] Body [user:display-name], A site administrator at [site:name] has created an account for you. You may now log in by clicking this link or copying and pasting it into your browser: [user:one-time-login-url] This link can only be used once to log in and will lead you to a page where you can set your password. After setting your password, you will be able to log in at [site:login-url] in the future using: username: [user:username] password: Your password -- [site:name] team
Welcome (awaiting approval)	
Admin (user awaiting approval)	
Welcome (no approval required)	
Account activation	
Account blocked	
Account cancellation confirmation	
Account canceled	
Password recovery	

5. Click *Save configuration* to save the changes.

Expand your understanding

- [Bagian 12.2, "Clearing the Cache"](#)
- [Bagian 7.4, "Creating a User Account"](#)

Related concepts

See [Bab 7, Mengelola Akun Pengguna](#) for more information about user accounts and permissions.

Additional resources

[Security Guide](#) can help you with a more safety-focused approach to configuration.

Attributions

Written and edited by [Laura Vass](#) at [Pronovix](#), and [Jojo Alphonso](#) at [Red Crackle](#).

4.6. Configuring the Theme

Goal

Edit the settings of the default core Bartik theme to change the color scheme and to add a logo.

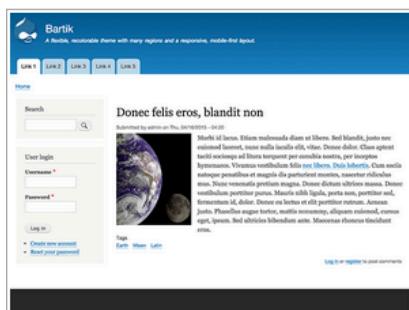
Prerequisite knowledge

[Bagian 1.3, "Concept: Themes"](#)

Steps

1. In the *Manage* administrative menu, navigate to *Appearance (admin/appearance)*.
2. Under *Installed themes*, you will find Bartik listed as your default theme. Under *Bartik (default theme)*, click *Settings*.

Installed themes



Bartik 8.2.0-dev (default theme)

A flexible, recolorable theme with many regions and a responsive, mobile-first layout.

Settings

3. Under *Color scheme*, click inside each color build box and type the proper color codes you would like to add. For example, use the following colors:

Area	Color
Header background top	#7db84a (green)
Header background bottom	#2a3524 (dark green)
Main background	#ffffff (white)
Sidebar background	#f8bc65 (light orange)
Sidebar borders	#e96b3c (orange)

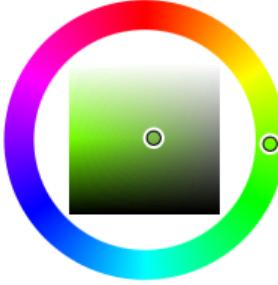
Area	Color
Footer background	#2a3524 (dark green)
Title and slogan	#ffffff (white)
Text color	#000000 (black)
Link color	#2a3524 (dark green)

Note: You can also use the color wheel on the right to select colors of your choice. The web color codes will be added for you.

▼ COLOR SCHEME

Color set Custom

Header background top		<input checked="" type="button"/> #7db84a
Header background bottom		<input checked="" type="button"/> #2a3524
Main background		<input checked="" type="button"/> #ffffff
Sidebar background		<input checked="" type="button"/> #f8bc65
Sidebar borders		<input checked="" type="button"/> #e96b3c
Footer background		<input checked="" type="button"/> #2a3524
Title and slogan		<input checked="" type="button"/> #ffffff
Text color		<input checked="" type="button"/> #000000
Link color		<input checked="" type="button"/> #2a3524



- Under *Logo image*, uncheck *Use the logo supplied by the theme*.

▼ LOGO IMAGE SETTINGS

Use the default logo supplied by the theme

Path to custom logo

 Examples: logo.svg (for a file in the public filesystem), public://logo.svg, or core/themes/bartik/logo.svg.

Upload logo image
 No file selected.

If you don't have direct file access to the server, use this field to upload your logo.

- Under *Upload logo image*, locate a logo file and upload it to your site. Note: You can also set a universal logo for all themes under *Appearance > Settings (admin/appearance/settings)*. A custom logo for your theme will override the universal logo.

Once you have selected the file you would like to upload, you will see its filename next to the *Choose File* or *Browse* button in your browser.

6. In order to save your changes and see the updated colors and logo on your site, click *Save configuration* at the bottom of the page.

Note: Under *Color scheme*, there is a *Preview* section that displays a sample of how your website will look with the new settings.

The screenshot shows the configuration preview for the Bartik theme. At the top, there's a green header bar featuring a circular logo with a sun, clouds, and a farm scene, surrounded by the text "ANYTOWN FARMERS MARKET" and "FARM FRESH FOOD". To the right of the logo, the word "Bartik" is displayed. Below the header is a navigation bar with three items: "Home", "Te Quidne", and "Vel Torqueo Quae Erat". The main content area has a white background with an orange rectangular callout box containing the text "Etiam est risus". To the right of this box is a large text block titled "Lorem ipsum dolor" followed by a paragraph of placeholder text. At the bottom, there's a dark footer section with a sidebar on the left containing the text "ETIAM EST RISUS" and a list of four items: "Donec placerat", "Nullam nibh dolor", "Blandit sed", and "Fermentum id".

7. Click *Return to site* or *Home* in the toolbar to verify that you have updated the core Bartik theme settings for your website.



Expand your understanding

- [*Bagian 11.4, "Finding Themes"*](#)
- [*Bagian 11.5, "Downloading and Installing a Theme from Drupal.org"*](#)
- If you do not see the effect of these changes in your site, you might need to clear the cache. See [*Bagian 12.2, "Clearing the Cache"*](#).

Attributions

Written and edited by [Ann Greazel](#), [Amanda Luker](#) at [Advomatic](#), and [Jack Haas](#).

Bab 5. Manajemen Dasar Halaman

5.1. Concept: Paths, Aliases, and URLs

What is a URL?

URL is the abbreviation for "Uniform Resource Locator", which is the page's address on the web. It is the "name" by which a browser identifies a page to display. In the example "Visit us at *example.com.*", *example.com* is the URL for the home page of your website. Users use URLs to locate content on the web.

What is a Path?

A path is the unique, last part of the URL for a specific function or piece of content. For example, for a page whose full URL is *http://example.com/node/7*, the path is *node/7*.

Here are some examples of paths you might find in your site:

- *node/7*
- *taxonomy/term/6*
- *admin/content/comment*
- *user/login*
- *user/3*

What is an Alias?

The core software has a feature called "URL Alias" that allows you to provide a more understandable name to the content. So, if you have an "About Us" page with the path *node/7*, you can set up an alias so that your visitors will see it as *http://www.example.com/AboutUs*. The core Path module, which supports URL aliasing, provides this functionality.

Related topics

- [*Bagian 5.2, "Creating a Content Item"*](#)
- [*Bagian 5.3, "Editing a Content Item"*](#)

Attributions

Adapted by [*Díána Lakatos*](#) at [*Pronovix*](#) from "[*URL aliases*](#)", and "[*Understanding Drupal paths*](#)" copyright 2000-2016 by the individual contributors to the [*Drupal Community Documentation*](#)

5.2. Creating a Content Item

Goal

Create and publish a content item that will be used as the home page of the site.

Prerequisite knowledge

[Bagian 5.1, "Concept: Paths, Aliases, and URLs"](#)

Site prerequisites

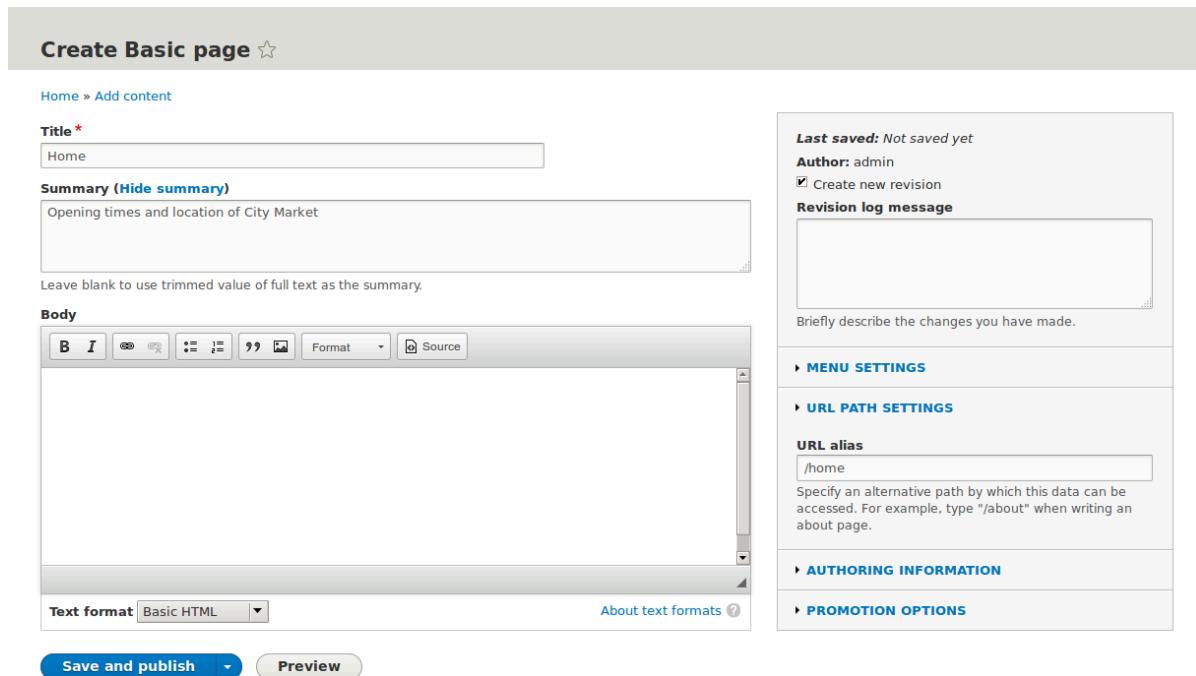
The *Basic page* content type must exist. This is created on your site when you install with the core Standard installation profile.

Steps

1. In the *Manage* administrative menu, navigate to *Content > Add content > Basic page (node/add/page)*. The *Create Basic page* form appears.
2. Click *Edit summary*.
3. Fill in the fields as shown below.

Field name	Explanation	Value
Title	Title of the page. Will also be used as a meta tag in the source code, URL alias, and as label of the content item in administration screens	Home
Summary	Summary of the value of the body field. Can be used as teaser in overview pages	Opening times and location of City Market.
Body	Full content of the page	Welcome to City Market - your neighborhood farmers market! Open: Sundays, 9 AM to 2 PM, April to September Location: Parking lot of Trust Bank, 1st & Union, downtown
URL path settings > URL aliases	Alternate relative path for the content	/home

By clicking the *Source* button in the rich text editor toolbar, you can see the HTML source code of the text that you are editing.



The screenshot shows the 'Create Basic page' interface in Drupal. On the left, there's a title field set to 'Home' and a summary field containing 'Opening times and location of City Market'. The body area has a rich text editor toolbar. On the right, there are various configuration sections: 'Last saved: Not saved yet', 'Author: admin', 'Create new revision' (checkbox checked), 'Revision log message' (text area), 'MENU SETTINGS', 'URL PATH SETTINGS' (with 'URL alias: /home'), 'AUTHORING INFORMATION', and 'PROMOTION OPTIONS'. At the bottom, there are 'Save and publish' and 'Preview' buttons.

4. Click *Preview* to ensure everything looks like expected.
5. Click *Back to content editing*.
6. Click *Save and publish*. The content is saved and can be found on the *Content* page.
7. Follow the same steps to create an About page, with title "About", and a body telling about the history of the farmer's market.

Expand your understanding

- [Bagian 5.5, "Designating a Front Page for your Site"](#)
- [Bagian 5.7, "Adding a Page to the Navigation"](#)
- [Bagian 10.4, "Translating Content"](#)

Related concepts

- [Bagian 10.1, "Concept: User Interface, Configuration, and Content translation"](#)
- [Bagian 5.1, "Concept: Paths, Aliases, and URLs"](#)
- [Bagian 5.3, "Editing a Content Item"](#)

Additional resources

[Drupal.org community documentation page "About nodes"](#)

Attributions

Written by [Agnes Kiss](#) and [Boris Doesborg](#).

5.3. Editing a Content Item

Goal

Update the hours on the Home page content item.

Prerequisite knowledge

[Bagian 5.2, "Creating a Content Item"](#)

Site prerequisites

A content item for the Home page must exist. See [Bagian 5.2, "Creating a Content Item"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Content* (`admin/content`).
2. If the content item you want to edit was updated or created recently, it should appear near the top of the content list on that page. If not, you can use the *Content type*, *Title*, or other filters to locate the content item.

The screenshot shows the 'Content' administration page. At the top, there are filters for 'Published status', 'Content type', 'Title', and 'Language'. Below these are buttons for 'Filter', 'With selection' (set to 'Delete content'), and 'Apply'. The main area displays a table of content items:

<input type="checkbox"/>	TITLE	CONTENT TYPE	AUTHOR	STATUS	UPDATED	OPERATIONS
<input type="checkbox"/>	About	Basic page	admin	Published	07/06/2016 - 15:55	<button>Edit</button>
<input type="checkbox"/>	Home	Basic page	admin	Published	07/06/2016 - 15:55	<button>Edit</button>

At the bottom left is an 'Apply' button.

3. Click *Edit* in the row of the content item you want to edit (Home), to open the content editing form. Update the opening hours in the *Body* field. See [Bagian 5.2, "Creating a Content Item"](#) for an explanation of the fields and a screenshot.
4. Check *Create new revision* in the box on the right, if it is not already checked, and enter a *Revision log message* explaining what changes you are making (for example, you might enter "Updated opening hours"). This text will appear in the revision log for the page.

Published

Last saved: 07/06/2016 - 15:55

Author: admin

Create new revision

Revision log message

Updated opening hours

Briefly describe the changes you have made.

5. Click *Save and keep published* to save your changes.
6. You will be redirected back to the *Content* administrative page, and there should be a message showing that the content item was updated.

✓ Basic page *Home* has been updated.

Expand your understanding

As an alternative to the first two steps above, you can also reach the content edit form as follows:

1. Starting from your site's home page, use the site's navigation menus to locate the page where the content you want to edit is displayed to visitors.
2. Most themes will display an *Edit* link or tab near the top of that page to people with permission to edit the page; clicking the link will take you to the full content edit form.
3. You could also use the in-place editor (refer to [Bagian 5.4, "Editing with the In-Place Editor"](#) from this page).

Attributions

Written by [Chris Dart](#) and [Jennifer Hodgdon](#).

5.4. Editing with the In-Place Editor

Goal

Use the in-place editor to add information to the About page without opening the full editor page.

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 5.2, "Creating a Content Item"](#)
- [Bagian 5.3, "Editing a Content Item"](#)
- [Bagian 4.1, "Concept: Administrative Overview"](#)

Site prerequisites

The About page you want to edit must exist. See [Bagian 5.2, "Creating a Content Item"](#).

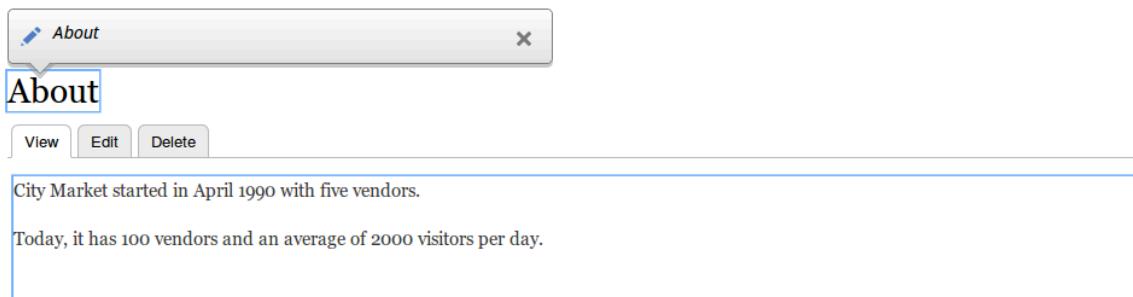
Steps

Navigate to the About page to edit content with the in-place editor. If the page has not yet been added to a navigation menu, you can find and open it by following these steps:

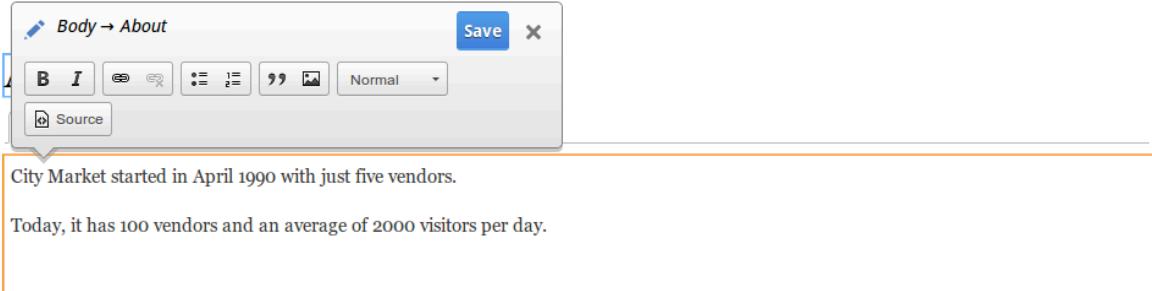
1. In the *Manage* administrative menu, navigate to *Content (admin/content)*.
2. If the content item you want to edit was updated or created recently, it should appear near the top of the content list on that page. If not, you can use *Type*, *Title*, or other filters to locate the content item.
3. Find the About page and click it.

Once the content you want to edit is being viewed in your browser, follow these steps to use the quick editor:

1. Find the section of the page that displays the content you want to edit, and turn on quick editing mode using the *Quick edit* contextual link (see [Bagian 4.1, "Concept: Administrative Overview"](#) for instructions on how to access contextual links). You will see each editable field on this content item outlined in blue; also, a hovering dialog box will appear.



2. Click the *Body* text area to begin editing. The *Body* field supports a rich text editor. The editing toolbar will be displayed in a hovering dialog box.
3. Add some information about City Market. A *Save* button will appear in the hovering dialog box, and the field outline will change color.



4. If you are satisfied with your edits, click *Save* in the hovering dialog box. If not, click "x" to discard your edits, and confirm. Either way, quick editing mode will be turned off.

Expand your understanding

Try using the full editor on the same content (see [Bagian 5.3, "Editing a Content Item"](#)) and note that there is much more information that can be edited that is not available in the quick editor.

Attributions

Written and edited by [David Lee](#) and [Jennifer Hodgdon](#).

5.5. Designating a Front Page for your Site

Goal

Configure which content item is displayed as the front page of your website.

Site prerequisites

The content item that you want to designate as the front page of your site must exist. See [Bagian 5.2, "Creating a Content Item"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Configuration > System > Basic site settings* (`admin/config/system/site-information`).
2. Under *Front page*, replace `/node` with the name of the page you would like to make the home page. To use the home page that was previously created, provide its path `/home`. Click *Save configuration*.

▼ FRONT PAGE

Default front page

http://example.com /home

Optionally, specify a relative URL to display as the front page. Leave blank to display the default front page.

3. Navigate to the home page to verify that it displays content as configured by you.

The screenshot shows a website for 'Anytown Farmers Market'. The header features a logo with a sun, clouds, and a tractor, and the text 'ANYTOWN FARMERS MARKET' and 'FARM FRESH FOOD'. The title 'Anytown Farmers Market' and subtitle 'Farm Fresh Food' are displayed. A navigation bar includes 'Home' (which is highlighted), 'Tools', and 'Add content'. Below the navigation, there are buttons for 'View', 'Edit', 'Delete', and 'Revisions'. The main content area welcomes visitors to the neighborhood farmers market, mentions open hours (Sundays, 9 AM to 2 PM, April to September), and provides the location (Parking lot of Trust Bank, 1st & Union, downtown).

Expand your understanding

- [Bagian 5.7, "Adding a Page to the Navigation"](#)
- Follow [Bagian 5.2, "Creating a Content Item"](#) to create an error page to be used as a 404 (page not found) or 403 (not authorized) response on your site. Then following the steps here, you can designate it as the error response, in the *Error pages* section of the configuration.

Related concepts

[Bagian 5.6, "Concept: Menu"](#)

Attributions

Written and edited by [Ann Greazel](#), [Jack Haas](#), and [Jojo Alphonso](#) at [Red Crackle](#).

5.6. Concept: Menu

What is a menu?

Menus are a collection of links (menu items) used to navigate a website. The core Menu UI module provides an interface to control and customize the menu system. Menus are

primarily displayed as a hierarchical list of links. By default, new menu items are placed inside a built-in menu labeled *Main navigation*, but administrators can also create custom menus.

The core Standard installation profile contains five menus:

Main navigation

Links to sections intended for site visitors. They are usually created by site administrators.

Administration

Links to administrative tasks. This menu mainly contains links supplied by modules on your site.

User account menu

Links to tasks associated with the user account such as *My account* and *Log out*.

Footer

Links to important pages within the site intended for the footer. They are usually created by site administrators.

Tools

Links to tasks necessary for site visitors. Some modules feature their links here.

You can customize menus in the following ways, using the menu administration functionality:

- Creating new custom menus.
- Adding new menu items.
- Reordering menu items by setting their "weight" or by dragging them into place.
- Renaming menu items.
- Changing the link title (the tooltip that appears when you mouse over a menu item).
- Moving a menu item into a different menu by editing its *Parent* property.

A menu item will only be shown to a visitor if they have the rights to view the page it links to. For example, the admin menu item is not shown to visitors who are not logged in.

Related topics

- [*Bagian 5.7, "Adding a Page to the Navigation"*](#)
- [*Bagian 5.8, "Changing the Order of Navigation"*](#)
- To display a menu, you will need to place the block that corresponds to the menu in a region of your theme; see [*Bagian 8.1, "Concept: Blocks"*](#), [*Bagian 2.1, "Concept: Regions in a Theme"*](#), and [*Bagian 8.3, "Placing a Block in a Region"*](#). The core Standard installation profile

places all of the menus it defines except Administration in regions of the core Bartik theme. The core Toolbar module, which is installed by the core Standard installation profile, displays the Administration menu; it is also displayed by the contributed Admin Toolbar module.

Attributions

Written and edited by [Ajay Viswambharan](#), [Joy Alphonso](#) at [Red Crackle](#), and [Jennifer Hodgdon](#).

5.7. Adding a Page to the Navigation

Goal

Add a page to the navigation. For example, the About page.

Prerequisite knowledge

- [Bagian 5.6, "Concept: Menu"](#)
- [Bagian 5.3, "Editing a Content Item"](#)

Site prerequisites

The About page content item must exist. See [Bagian 5.2, "Creating a Content Item"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Content* (`admin/content`).
2. Find the About page, and click *Edit* in that row. The content editing form appears.

<input type="checkbox"/>	TITLE	CONTENT TYPE	AUTHOR	STATUS	UPDATED	OPERATIONS
<input type="checkbox"/>	Home	Basic page	admin	Published	07/06/2016 - 12:52	<button>Edit</button> <button>⋮</button>
<input type="checkbox"/>	About	Basic page	admin	Published	07/06/2016 - 12:52	<button>Edit</button> <button>⋮</button>

3. Click *Menu settings* on the right to expand it.
4. Check *Provide a menu link* for the menu options to appear.
5. Enter values from the table below:

Field name	Explanation	Example value
Menu link title	Title that will be displayed in the menu	About

Field name	Explanation	Example value
Description	Text that will be displayed when a visitor hovers over the link	History of the market
Parent item	Location of the page in the menu hierarchy. For example, if you choose <Main navigation>, the page will appear in the highest level of the navigation. By choosing another menu item as parent, you can create a menu hierarchy of multiple levels.	<Main navigation>
Weight	The order in which the page should appear in the menu (lower-weighted menu items will be shown before higher-weighted menu items)	-2

▼ MENU SETTINGS (ABOUT)

Provide a menu link

Menu link title
About

Description
History of the market

Shown when hovering over the menu link.

Parent item
<Main navigation> ▾

Weight
-2

Menu links with lower weights are displayed before links with higher weights.

6. Click *Save and keep published* to save the changes. Click *Home* or *Return to site* in the navigation bar to see the result, which could look like the picture below.

My account Log out

ANYTOWN FARMERS MARKET

FARM FRESH FOOD

About Home

Tools

Add content

Home

View Edit Delete Revisions

Welcome to City Market - your neighborhood farmers market!

Open: Sundays, 9 AM to 2 PM, April to September

Location: Parking lot of Trust Bank, 1st & Union, downtown

Expand your understanding

[Bagian 5.8, "Changing the Order of Navigation"](#)

Additional resources

[Drupal.org community documentation page "Working with menus"](#)

Attributions

Adapted by [Boris Doesborg](#) from ["Working with Menus"](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#); edited by [Jack Haas](#).

5.8. Changing the Order of Navigation

Goal

Reorder the items in a menu.

Prerequisite knowledge

- [Bagian 5.6, "Concept: Menu"](#)
- [Bagian 5.7, "Adding a Page to the Navigation"](#)

Site prerequisites

Home and About pages must exist in the main navigation menu. See [Bagian 5.7, "Adding a Page to the Navigation"](#).

Steps

- In the *Manage* administrative menu, navigate to *Structure > Menus (admin/structure/menu)* where all menus on your site are listed. Click *Edit Menu* from the *Operations* dropdown for *Main navigation*. You can also reach this page using contextual links (refer to [Babian 4.1, "Concept: Administrative Overview"](#)) for the menu.

TITLE	DESCRIPTION	OPERATIONS
Administration	Administrative task links	Edit menu ▾
Footer	Site information links	Edit menu ▾
Main navigation	Site section links	Edit menu ▾
Tools	User tool links, often added by modules	Edit menu ▾
User account menu	Links related to the active user account	Edit menu ▾

- The *Edit menu* page will display a list of each item in the menu you chose (*Main navigation*).

MENU LINK	ENABLED	OPERATIONS
⊕ About	<input checked="" type="checkbox"/>	Edit ▾
⊕ Home	<input checked="" type="checkbox"/>	Edit
Save		

- Use the cross bar handles to reorder the menu items. Drag the Home menu item above the About menu item so that it appears first.

⚠ *You have unsaved changes.

MENU LINK	ENABLED	OPERATIONS
⊕ Home	<input checked="" type="checkbox"/>	Edit
⊕ About	<input checked="" type="checkbox"/>	Edit ▾
Save		

- Click *Save*.
- The home page now displays the main navigation with the Home menu item displayed first.



Expand your understanding

Add a menu item called *Contact*, leading to the `/contact` page, to your Main navigation menu. The contact page is provided by the core Contact module; you may want to edit its layout and fields (see [Bagian 6.9, "Changing Content Entry Forms"](#)).

Related concepts

[Bagian 5.6, "Concept: Menu"](#)

Attributions

Written by [Ann Greazel](#).

Bab 6. Menyiapkan Struktur Konten

6.1. Adding a Content Type

Goal

Add and configure a new content type Vendor.

Prerequisite knowledge

[Bagian 2.3, "Concept: Content Entities and Fields"](#)

Site prerequisites

You need to have a plan in place for your content structure. See [Bagian 2.5, "Planning your Content Structure"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Content types (admin/structure/types)*. The *Content types* page appears showing all the available types of content.
2. Click *Add content type*. The *Add content type* page appears. Fill in the fields as shown below.

Field name	Explanation	Example value
Name	Name of the content type	Vendor
Description	Explain the use of the content type	Information about a vendor

[Home](#) » [Administration](#) » [Structure](#) » [Content types](#)

Individual content types can have different fields, behaviors, and permissions assigned to them.

Name *

Machine name: vendor [\[Edit\]](#)

The human-readable name of this content type. This text will be displayed as part of the list on the *Add content* page. This name must be unique.

Description

This text will be displayed on the *Add new content* page.

3. In the vertical tab *Submission form settings*, configure the form that is used for creating and editing content of this type. Fill in the fields as shown below.

Field name	Explanation	Example value
Title field label	Label of the Title field that is shown when editing or creating content of this type.	Vendor name
Preview before submitting	Option to choose whether you should preview the content before submitting.	Optional
Explanation or submission guidelines	Instructions for creating or editing content.	(Leave blank)

Submission form settings

Vendor name

Publishing options
Published , Create new revision

Display settings
Don't display post information

Menu settings

Title field label *
Vendor name

Preview before submitting
 Disabled
 Optional
 Required

Explanation or submission guidelines

This text will be displayed at the top of the page when creating or editing content of this type.

4. In the vertical tab *Publishing options*, decide on default options for new content of this type. Fill in the fields as shown below.

Field name	Explanation	Example value
Published	Make the content item published by default.	Checked
Promoted to front page	In a default website, this setting can be used to show content on the homepage.	Unchecked
Sticky at top of lists	In a default website, this setting can be used to keep content on top of a list.	Unchecked

Field name	Explanation	Example value
Create new revision	Create a new revision each time the vendor is being edited.	Checked

Changing these settings does not affect the content items that have already been created.

The screenshot shows the 'Vendor' content type configuration page. On the left, there is a vertical tab menu with four sections: 'Submission form settings', 'Publishing options', 'Display settings', and 'Menu settings'. The 'Publishing options' section is expanded, showing the 'Create new revision' checkbox is checked. To the right of this section, there is a summary of the checked options: 'Default options' (Published, Create new revision checked), and a note that users with the 'Administer content' permission can override these settings.

5. In the vertical tab *Display settings*, decide if the author and publication date will be visible in the content item. Fill in the fields as shown below.

Field name	Explanation	Example value
Display author and date information	Display the author username and publication date on each vendor page.	Unchecked

The screenshot shows the 'Vendor' content type configuration page. The 'Display settings' section is expanded, showing the 'Display author and date information' checkbox is unchecked. A note next to the checkbox states: 'Author username and publish date will be displayed.'

6. In the vertical tab *Menu settings*, fill in the fields as shown below.

Field name	Explanation	Example value
Available menus	Menus that this type of content can be added to. Vendors do not need to appear in menus, so uncheck all menu options.	Unchecked

The screenshot shows the 'Vendor' content type configuration page. On the left, there are four sections: 'Submission form settings' (Vendor name), 'Publishing options' (Published, Create new revision), 'Display settings' (Don't display post information), and 'Menu settings'. On the right, under 'Available menus', there is a list of menu types: Administration, Footer, Main navigation, Tools, and User account menu. Below this list is a note: 'The menus available to place links in for this content type.'

- Click *Save and manage fields* to save the content type. The *Manage fields* page appears that allows you to add fields to the content type. See [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)

The screenshot shows the 'Manage fields' page for the 'Vendor' content type. At the top, there are tabs: Edit, Manage fields (which is active), Manage form display, and Manage display. Below the tabs, a breadcrumb trail shows: Home > Administration > Structure > Content types > Vendor. A green success message box contains the text: '✓ The content type Vendor has been added.' Below the message is a button labeled '+ Add field'. A table lists the fields: Body (Machine Name: body, Field Type: Text (formatted, long, with summary), Operations: Edit button). The table has columns: LABEL, MACHINE NAME, FIELD TYPE, and OPERATIONS.

- Follow the same steps to create a content type for recipes. Example values for the fields in the forms, where they are different from the steps above:

Field name	Example value
Name	Recipe
Description	A recipe submitted by a vendor
Submission form settings - Title	Recipe name

Expand your understanding

- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)
- Install and configure the [contributed Pathauto module](#) so that content items get automatically generated URLs/path aliases. See [Bagian 5.1, "Concept: Paths, Aliases, and URLs"](#) for more on URLs within your site, [Bagian 11.1, "Finding Modules"](#) for instructions on finding contributed modules, and [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#) for instructions on downloading and installing contributed modules.

Attributions

Written and edited by [Sree Veturi](#), [Boris Doesborg](#), and [Jennifer Hodgdon](#).

6.2. Deleting a Content Type

Goal

Delete the unneeded content type *Article*.

Prerequisite knowledge

[Bagian 4.1, "Concept: Administrative Overview"](#)

Site prerequisites

The *Article* content type must exist. It is created on your site when you install with the core Standard installation profile.

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Content types* ([admin/structure/types](#)). The *Content types* page appears.
2. Click *Delete* in the *Operations* dropdown button for the *Article* content type.

The screenshot shows the 'Content types' administration page. At the top, there's a header bar with the title 'Content types' and a star icon. Below it is a breadcrumb trail: 'Home > Administration > Structure'. A blue button labeled '+ Add content type' is visible. The main area has a table with columns: 'NAME', 'DESCRIPTION', and 'OPERATIONS'. The table contains four rows:

NAME	DESCRIPTION	OPERATIONS
Article	Use <i>articles</i> for time-sensitive content like news, press releases or blog posts.	Manage fields Manage form display Manage display Edit Delete
Basic page	Use <i>basic pages</i> for your static content, such as an 'About us' page.	
Recipe	Recipe submitted by a vendor	
Vendor	Information about a vendor	Manage fields

A dropdown menu for 'Manage fields' is open over the last row. The third item in the list, 'Delete', is highlighted with a light gray background. The URL for the page is [admin/structure/types/manage/article/delete](#).

3. A confirmation page is displayed. Click *Delete*.

Are you sure you want to delete the content type Article?



[Home](#) » [Administration](#) » [Structure](#) » [Content types](#) » Article

This action cannot be undone.

▼ CONFIGURATION DELETIONS

The listed configuration will be deleted.

Entity form display

- node.article.default

Entity view display

- node.article.default
- node.article.rss
- node.article.teaser

Field

- Body
- Comments
- Image
- Tags

RDF mapping

- node.article

[Delete](#)

[Cancel](#)

4. The *Content types* page appears with a confirmation message saying that the content type has been deleted:

The content type Article has been deleted.

Attributions

Written and edited by [Sree Veturi](#) and [Boris Doesborg](#).

6.3. Adding Basic Fields to a Content Type

Goal

Add a link field and an image field to the Vendor content type.

Prerequisite knowledge

[Bagian 2.3, "Concept: Content Entities and Fields"](#)

Site prerequisites

The Vendor content type must exist. See [Bagian 6.1, "Adding a Content Type"](#).

Steps

Add the fields Vendor URL and Main image to the Vendor content type.

1. In the *Manage* administrative menu, navigate to *Structure > Content types (admin/structure/types)*. Then click *Manage fields* in the dropdown button for the Vendor content type. The *Manage fields* page appears.
2. Click *Add field*. The *Add field* page appears. You can either create a new field for the content type or re-use an existing field.
3. Fill in the fields as shown below.

Field name	Explanation	Value
Add a new field	Field type	Link
Label	Label that is visible in administration pages	Vendor URL

A machine name is automatically generated, based on the *Label* value. Click *Edit* if you want to override the default name.

4. Click *Save and continue*. The page *Vendor URL* appears which lets you set the allowed number of values. Fill in the fields as shown below.

Field name	Explanation	Value
Allowed number of values	The number of values that can be entered	Limited, 1

5. Click *Save field settings*. The page *Vendor URL settings for Basic page* appears which allows you to configure the field. Fill in the fields as shown below.

Field name	Explanation	Value
Label	Label that is visible in the content form	Vendor URL
Help text	The instruction that is shown below the field	(leave blank)
Required field	Whether the field is required or not	Unchecked
Allowed link type	The kind of links that can be entered	External links only
Allow link text	Whether a link text can be entered	Disabled

Vendor URL settings for Vendor ★

[Edit](#) [Field settings](#)

Home » Administration » Structure » Content types » Vendor » Manage fields

Label*
Vendor URL

Help text

Instructions to present to the user below this field on the editing form.
 Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

 This field supports tokens.

Required field

▼ DEFAULT VALUE
 The default value for this field, used when creating new content.
Vendor URL
 This must be an external URL such as <http://example.com>.

Allowed link type
 Internal links only
 External links only
 Both internal and external links

Allow link text
 Disabled
 Optional
 Required

- Click *Save settings*. The Vendor URL has been added to the content type. Continue creating the Main image field.
- Click *Add field*. The *Add field* page appears. Fill in the fields as shown below.

Field name	Explanation	Value
Add a new field	Field type	Image
Label	Label that is visible in administration pages	Main image

- Click *Save and continue*. The page Main image appears. Fill in the fields as shown below.

Field name	Explanation	Value
Allowed number of values	The number of values that can be entered	Limited, 1

You can set a default image here. This will be used when you do not provide an image when creating a Vendor content item.

- Click *Save field settings*. The page *Main image settings for Basic page* appears. Fill in the fields as shown below.

Field name	Explanation	Value
Label	Label that is visible in the content form	Main image
Help text	The instruction that is shown below the field	(leave blank)
Required field	Whether the field is required or not	Checked
Allowed file extensions	The type of images that can be uploaded	png, gif, jpg, jpeg
File directory	The directory where the files will be stored. By providing a file directory value, you ensure that all images uploaded via the Main image field will be located in the same directory.	vendors
Minimum image resolution	The minimum resolution of the uploaded image	600 x 600
Maximum upload size	The maximum file size of the uploaded image	5 MB
Enable Alt field	Whether an alternative text can be entered	Checked
Alt field required	Whether an alternative text is required	Checked

Menyiapkan Struktur Konten

Main image settings for Vendor ☆

[Edit](#) [Field settings](#)

Home » Administration » Structure » Content types » Vendor » Manage fields

Label*
Main image

Help text

Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

This field supports tokens.

Required field

DEFAULT IMAGE

If no image is uploaded, this image will be shown on display and will override the field's default image.

Image
[Browse...](#) No file selected.

Image to be shown if no image is uploaded.

Alternative text

This text will be used by screen readers, search engines, and when the image cannot be loaded.

Title

The title attribute is used as a tooltip when the mouse hovers over the image.

Allowed file extensions *
png, gif, jpg, jpeg

Separate extensions with a space or comma and do not include the leading dot.

File directory
vendors

10. Click *Save settings*. Main image has been added to the content type.

Manage fields ☆

[Edit](#) [Manage fields](#) [Manage form display](#) [Manage display](#)

Home » Administration » Structure » Content types » Vendor

✓ Saved Main image configuration.

[+ Add field](#)

LABEL	MACHINE NAME	FIELD TYPE	OPERATIONS
Body	body	Text (formatted, long, with summary)	Edit ▾
Main image	field_main_image	Image	Edit ▾
Vendor URL	field_vendor_url	Link	Edit ▾

11. Add a Main image field to the Recipe content type, using similar steps. Start by navigating to the Recipe content type's *Manage Fields* page in step 1. Then skip to step 7 and follow the remaining steps, but reuse the existing Main image field you created for the Vendor content type rather than creating a new field. In subsequent steps, some of the configuration screens will not be available, because of the field reuse.
12. Create two Vendor content items (see [Bagian 5.2, "Creating a Content Item"](#)) called "Happy Farm" and "Sweet Honey". Make sure that they include images and URLs.

Expand your understanding

- [Bagian 6.12, "Concept: Image Styles"](#)
- [Bagian 6.11, "Changing Content Display"](#)
- [Bagian 6.9, "Changing Content Entry Forms"](#)

Additional resources

[Drupal.org community documentation page "Add a field to a content type"](#)

Attributions

Written by [Sree Veturi](#) and [Boris Doesborg](#).

6.4. Concept: Reference Fields

Prerequisite knowledge

[Bagian 2.3, "Concept: Content Entities and Fields"](#)

What is a reference field?

A *reference field* is a field that represents a relationship between an entity and one or more other entities, which may belong to the same or different entity type. The three most commonly-used reference fields are:

Content reference

A reference to a content item. For example, you might want to connect recipes to the vendors who submitted them. You would set up a content reference field called Submitted by referencing Vendor content items on the Recipe content type.

Taxonomy term reference

A reference to a taxonomy term. For example, you might want to connect recipes to their ingredients. You would set up a taxonomy term reference field called Ingredients on the Recipe content type. This reference field will point to the vocabulary Ingredients.

User reference

A reference to a user account. For example, you might want to connect recipes with their chefs. You would set up a user reference field called Chefs on the Recipe content type.

Related topics

[Bagian 6.5, "Concept: Taxonomy"](#)

Attributions

Written and edited by [Surendra Mohan](#), and [Jojo Alphonso](#) at [Red Crackle](#).

6.5. Concept: Taxonomy

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 6.4, "Concept: Reference Fields"](#)

What is Taxonomy?

Taxonomy is used to classify website content. One common example of taxonomy is the tags used to classify or categorize posts in a blog website; the farmers market website could use an ingredients taxonomy to classify recipes. Individual taxonomy items are known as *terms* (the blog tags or recipe ingredients in these examples); and a set of terms is known as a *vocabulary* (the set of all blog post tags, or the set of all recipe ingredients in these examples). Technically, taxonomy terms are an entity type and the entity subtypes are the vocabularies. Like other entities, taxonomy terms can have fields attached; for instance, you could set up an image field to contain an icon for each term.

An individual vocabulary can organize its terms in a hierarchy, or it could be flat. For example, blog tags normally have a flat structure, while a recipe ingredients vocabulary could be hierarchical (for example, tomatoes could be a sub-term of vegetables, and under tomatoes, you could have green and red tomatoes).

Taxonomy terms are normally attached as reference fields to other content entities, which is how you can use them to classify content. When you set up a taxonomy reference field, you can let users enter terms in two ways:

Free tagging

New terms can be created right on the content editing form.

Fixed list of terms

The list of terms is curated and managed outside the content editing form, and users can only choose from the existing list when editing content.

Taxonomy reference fields can be added to any entity, such as user accounts, custom blocks, or regular content items. If you use them to classify regular content items, your site will automatically be set up with taxonomy listing pages for each term; each of these pages lists all of the content items that are classified with that term. For example, if you created several recipes that all had carrots as an ingredient, you might see something like this on the Carrots taxonomy listing page:

Carrots

Fresh Carrots

Serve multi-colored carrots on a plate for dinner.

[Read more](#)

Green Salad

Chop up your favorite vegetables and put them in a bowl.

[Read more](#)

Related topics

- [Bagian 6.6, "Setting Up a Taxonomy"](#).
- The listing pages are views, which are covered in [Bab 9, Membuat Daftar dengan Views](#).

Attributions

Adapted and edited by [Surendra Mohan](#), [Jennifer Hodgdon](#), and [Jojo Alphonso](#) at [Red Crackle](#) from "[Organizing content with taxonomies](#)" and "[About taxonomies](#)", copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

6.6. Setting Up a Taxonomy

Goal

Create an Ingredients vocabulary and add it to the Recipe content type as a field that can contain an unlimited number of values and that allows adding new terms to the vocabulary.

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 6.5, "Concept: Taxonomy"](#)
- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)

Site prerequisites

The Recipe content type must exist. See [Bagian 6.1, "Adding a Content Type"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Taxonomy (admin/structure/taxonomy)*. You will see the *Tags* vocabulary that was created with the core Standard installation profile.

The screenshot shows the 'Taxonomy' page with a title bar 'Taxonomy ★'. Below it is a breadcrumb trail 'Home » Administration » Structure'. A descriptive text states: 'Taxonomy is for categorizing content. Terms are grouped into vocabularies. For example, a vocabulary called "Fruit" would contain the terms "Apple" and "Banana".' A blue button '+ Add vocabulary' is visible. The main area has a table with two columns: 'VOCABULARY NAME' and 'OPERATIONS'. One row is shown for 'Tags', with an 'Operations' dropdown set to 'List terms'.

2. Click *Add vocabulary*, and fill in the values below.
- | Field name | Explanation | Example value |
|-------------|-----------------------------------|---------------|
| Name | The name of the vocabulary | Ingredients |
| Description | A brief note about the vocabulary | (Leave blank) |

The screenshot shows the 'Add vocabulary' page for the 'Ingredients' vocabulary. The title bar says 'Add vocabulary ★'. The breadcrumb trail is 'Home » Administration » Structure » Taxonomy'. It has fields for 'Name*' (set to 'Ingredients') and 'Description'. A large blue 'Save' button is at the bottom.

3. Click *Save*. You will be taken to the *Ingredients* page, which shows a list of all the terms in this vocabulary.

The screenshot shows the 'Ingredients' vocabulary management page in Drupal. At the top, there is a navigation bar with links: 'List', 'Edit', 'Manage fields', 'Manage form display', and 'Manage display'. Below the navigation, a breadcrumb trail shows the path: 'Home > Administration > Structure > Taxonomy > Ingredients'. A success message in a green box states: '✓ Created new vocabulary *Ingredients*.'. A note below it says: 'You can reorganize the terms in *Ingredients* using their drag-and-drop handles, and group terms under a parent term by sliding them under and to the right of the parent.' A blue button labeled '+ Add term' is visible. To the right, there is a link 'Show row weights'. The main area has a table with three columns: 'NAME', 'WEIGHT', and 'OPERATIONS'. A message at the bottom of the table says: 'No terms available. [Add term](#)'.

4. Click *Add term*. Enter "Butter" in the *Name* field. Click *Save*.

Add term ☆

Home » Administration » Structure » Taxonomy » Ingredients

Name *
Butter
The term name.

Description

Text format Basic HTML ▾ [About text formats](#) ?

A description of the term.

RELATIONS

URL alias

Specify an alternative path by which this data can be accessed. For example, type "/about" when writing an about page.

Save

5. You will receive a confirmation about the term you created. Add more terms. For example, "Eggs" and "Milk".
6. In the *Manage* administrative menu, navigate to *Structure > Content Types* (`admin/structure/types`). Click *Manage fields* for your Recipe content type.
7. Click *Add field*, and enter values from the table below. Click *Save and continue*.

Field name	Explanation	Value
Add a new field	Select the field type	Reference > Taxonomy term
Label	The title to give the field	Ingredients

Add field ☆

Home » Administration » Structure » Content types » Recipe » Manage fields

Add a new field or **Re-use an existing field**

Label *
Ingredients Machine name: field_ingredients [Edit]

Save and continue

8. On the following configuration screen, enter the values from the table below. Click *Save field settings*.

Field name	Explanation	Value
Type of item to reference	The type of entity that is referenced by the field	Taxonomy term
Allowed number of values	The number of values a user can enter	Unlimited

These settings apply to the *Ingredients* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

Type of item to reference *
Taxonomy term

Allowed number of values
Unlimited

Save field settings

9. On the following configuration screen, enter the values from the table below. Click *Save settings*.

Field name	Explanation	Value
Help text	Help shown to users creating content	Enter ingredients that site visitors might want to search for
Reference type > Reference method	Select the method used to choose allowed values	Default
Reference type > Available Vocabularies	Select the vocabulary to choose allowed values from	Ingredients
Reference type > Create referenced entities if they don't already exist	Whether new ingredient terms can be created from the content editing form	Checked

Menyiapkan Struktur Konten

Label*
Ingredients

Help text
Enter ingredients that site visitors might want to search for

Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

This field supports tokens.

Required field

DEFAULT VALUE

REFERENCE TYPE

Reference method*
Default

Create referenced entities if they don't already exist

Available Vocabularies*

Ingredients
 Tags

[Save settings](#) [Delete](#)

10. Click *Save settings*. You will be taken back to the *Manage Fields* page. A message will be displayed saying that the configuration for Ingredients is complete.

Manage fields ☆

Edit Manage fields Manage form display Manage display

Home » Administration » Structure » Content types » Recipe

✓ Saved Ingredients configuration.

[+ Add field](#)

LABEL	MACHINE NAME	FIELD TYPE	OPERATIONS
Body	body	Text (formatted, long, with summary)	Edit ▾
Ingredients	field_ingredients	Entity reference	Edit ▾
Main image	field_main_image	Image	Edit ▾

Attributions

Written and edited by [Bob Snodgrass](#), and [Joy Alphonso](#) at [Red Crackle](#).

6.7. Adding a Reference Field

Goal

Add a reference field so that recipes can be linked to the vendor that has submitted it.

Prerequisite knowledge

- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)
- [Bagian 6.4, "Concept: Reference Fields"](#)
- [Bagian 6.1, "Adding a Content Type"](#)

Site prerequisites

The Recipe and Vendor content types must exist. See [Bagian 6.1, "Adding a Content Type"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Content types (admin/structure/types)*. Then click *Manage fields* in the dropdown button for the Recipe content type. The *Manage fields* page appears.
2. Click *Add field*. The *Add field* page appears. Fill in the fields as shown below. Click *Save and continue*.

Field name	Explanation	Value
Add a new field	Option to specify the field type	Reference > Content
Label	The title you want to give the field	Submitted by

Home > Administration > Structure > Content types > Recipe > Manage fields

Add a new field **Re-use an existing field**

Content or - Select an existing field -

Label*
Submitted by Machine name: field_submitted_by [Edit]

Save and continue

3. The page *Submitted by* appears which lets you set the allowed number of values. Fill in the fields as shown below. Click *Save field settings*.

Field name	Explanation	Value
Type of item to reference	Option to select the type of referenced item	Content
Allowed number of values	Specify the count of values associated with the field	Limited, 1

Home » Administration » Structure » Content types » Recipe » Manage fields » Submitted by

These settings apply to the *Submitted by* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

Type of item to reference *

Content

Allowed number of values

Limited 1

Save field settings

- The page *Submitted by settings for Recipe* appears which allows you to configure the field. Fill in the fields as shown below. Click *Save settings*.

Field name	Explanation	Value
Label	Title shown for this field on the page	Submitted by
Help text	Brief text aiding the person creating content	Choose the vendor that submitted this recipe
Required field	Whether a value has to be provided or not	Checked
Reference method	Option to select reference method	Default
Content types	Specify the content type	Vendor
Sort by	Sorting field	Title
Sort direction	Sorting order	Ascending

Menyiapkan Struktur Konten

Help text
Choose the vendor that submitted this recipe

Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

This field supports tokens.

Required field

DEFAULT VALUE
The default value for this field, used when creating new content.
Submitted by

REFERENCE TYPE

Reference method *
Default

Create referenced entities if they don't already exist

Content types *
 Basic page
 Recipe
 Vendor

Sort by
Vendor name

Sort direction *
Ascending

Save settings [Delete](#)

5. The Submitted by field has been added to the content type.

Manage fields ☆

Edit Manage fields Manage form display Manage display

Home » Administration » Structure » Content types » Recipe

✓ Saved Submitted by configuration.

+ Add field

LABEL	MACHINE NAME	FIELD TYPE	OPERATIONS
Body	body	Text (formatted, long, with summary)	Edit ▾
Ingredients	field_ingredients	Entity reference	Edit ▾
Main image	field_main_image	Image	Edit ▾
Submitted by	field_submitted_by	Entity reference	Edit ▾

Attributions

Written and edited by [Boris Doesborg](#), and [Jojo Alphonso](#) at [Red Crackle](#).

6.8. Concept: Forms and Widgets

Prerequisite knowledge

[Bagian 2.3, "Concept: Content Entities and Fields"](#)

What are forms and widgets?

The content management system software that your site is running allows administrators to edit content and configure settings online, using various web *forms*. In particular, *content editing forms* are used to edit your site's content, and they are configurable by administrators; settings configuration forms are provided by modules and cannot themselves be configured.

The data in your site's content is stored in one or more fields that are attached to the content type and/or sub-type. When you configure the content editing form for each content sub-type, you can:

- Select a *widget* for each field. A widget defines the method used to enter the data for the field. For example, a taxonomy term can be chosen using an autocomplete field, a select list, or a tags-style field that lets editors add new tags automatically.
- Configure widget settings. For example, you can choose the size of a plain-text entry field.
- Hide one or more fields from the editing form.
- Reorder the fields.

In principle, you can also have multiple content editing forms available for each content sub-type. This feature is rarely used, however; the only exception in common use is for the user profile fields: you can use different forms for user registration and user editing. For example, you might have a limited set of fields shown when users first register on the site, and more fields shown later on when they edit their profiles.

Related topics

- [Bagian 6.9, "Changing Content Entry Forms"](#)
- [Bagian 6.10, "Concept: View Modes and Formatters"](#)

Attributions

Written by [Jennifer Hodgdon](#).

6.9. Changing Content Entry Forms

Goal

Change the Recipe form to use a different widget to enter terms in the Ingredients field.

Prerequisite knowledge

- [Bagian 6.1, "Adding a Content Type"](#)
- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)
- [Bagian 6.5, "Concept: Taxonomy"](#)
- [Bagian 6.8, "Concept: Forms and Widgets"](#)

Site prerequisites

The Recipe content type must exist, and it must have an Ingredients taxonomy term reference field. See [Bagian 6.1, "Adding a Content Type"](#) and [Bagian 6.6, "Setting Up a Taxonomy"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Content > Add content > Recipe (node/add/recipe)* to look at the content entry form that is set up by default. Notice how you have to enter ingredients one by one, instead of having a more compact format.
2. In the *Manage* administrative menu, navigate to *Structure > Content types (admin/structure/types)*. Then click *Manage form display* on the dropdown button for the Recipe content type. The *Manage form display* page appears.
3. For the Ingredients field, select *Autocomplete (Tags style)* in the *Widget* column.

Menyiapkan Struktur Konten

FIELD	WIDGET	
✚ Recipe name	Textfield ▾	Textfield size: 60
✚ Authored by	Autocomplete ▾	Autocomplete matching: Contains Textfield size: 60 No placeholder
✚ Authored on	Datetime Timestamp ▾	
✚ Promoted to front page	Single on/off checkbox ▾	Use field label: Yes
✚ Sticky at top of lists	Single on/off checkbox ▾	Use field label: Yes
✚ URL alias	URL alias ▾	
✚ Body	Text area with a summary ▾	Number of rows: 9 Number of summary rows: 3
✚ Main image	Image ▾	Preview image style: Thumbnail (100x100) Progress indicator: throbber
✚ Ingredients	Autocomplete (Tags style) ▾	Autocomplete matching: Contains Textfield size: 60 No placeholder
✚ Submitted by	Autocomplete ▾	Autocomplete matching: Contains Textfield size: 60 No placeholder
Disabled		
No field is hidden.		
Save		

4. Click **Save**.
5. In the *Manage* administrative menu, navigate to *Content > Add content > Recipe (node/add/recipe)* to verify the changed behavior of the content form. The Ingredients field is now a single text field that accepts multiple values.

Create Recipe ☆

Home » Node » Add content

Recipe name *

Body (Edit summary)

Main image *

Ingredients

Submitted by *

Text format Basic HTML [About text formats](#)

Last saved: Not saved yet
Author: admin
 Create new revision
Revision log message
 Briefly describe the changes you have made.

URL PATH SETTINGS
 AUTHORIZING INFORMATION
 PROMOTION OPTIONS

6. Create two Recipe content items (see [Bagian 5.2, "Creating a Content Item"](#)), such as recipes for "Green Salad" and "Fresh Carrots". Make sure all the fields have values, including images, ingredients, and submitted by (set this to one of the Vendor content items you created in [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)).

Expand your understanding

Change the main site Contact form by navigating in the *Manage* administrative menu to *Structure > Contact forms*. For instance, you may want to hide the *Send yourself a copy* or *Language* fields.

Attributions

Written by [Boris Doesborg](#).

6.10. Concept: View Modes and Formatters

Prerequisite knowledge

[Bagian 6.1, "Adding a Content Type"](#)

What is a View mode?

How an entity (such as content, a user or a comment) is displayed, depends on the context in which it is shown. This context is known as a *view mode*. Examples of view modes are:

- a full page of the content with all the field values
- a teaser of the content with a thumbnail image and a *Read more* link
- a full user profile with a zoomable portrait photo
- a user avatar with username and a link to the profile

Every view mode can be configured through the administration pages. To see the configuration options for the view modes of the Recipe content type, navigate in the *Manage* administrative menu to *Structure > Content types* (`admin/structure/types`) and click *Manage display* from the dropdown button. In each view mode, all fields can be hidden or displayed, and if they are displayed, you can choose and configure the field formatter.

What is a field formatter?

A field formatter is a setting for displaying the field values. For example, long text fields can be displayed trimmed or full-length, and taxonomy term reference fields can be displayed in plain text or linked to the taxonomy term page. Consult the *Manage display* page to see the field formatters for the fields of the Recipe content type.

Related topics

- [*Bagian 6.1, "Adding a Content Type"*](#)
- [*Bagian 6.11, "Changing Content Display"*](#)

Additional resources

[Drupal.org community documentation page "View modes"](#)

Attributions

Adapted by [*Boris Doesborg*](#) from [*"View modes"*](#), copyright 2000-2016 by the individual contributors to the [*Drupal Community Documentation*](#).

6.11. Changing Content Display

Goal

Make the content items more readable, accessible, and visibly attractive by reordering the fields, hiding labels, and tuning the output of the fields.

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 6.10, "Concept: View Modes and Formatters"](#)

Site prerequisites

The Vendor content type must exist, it must have Main Image and Vendor URL fields, and your site must have at least one Vendor content item. See [Bagian 6.1, "Adding a Content Type"](#), [Bagian 6.3, "Adding Basic Fields to a Content Type"](#), and [Bagian 5.2, "Creating a Content Item"](#).

Steps

1. Find and view a Vendor content item you created in [Bagian 6.3, "Adding Basic Fields to a Content Type"](#). Notice that there are several things that could be done to improve how the page looks:
 - The Main Image and Vendor URL fields should not have labels.
 - The order of the fields should be changed so that the image comes first.
 - The image should be smaller.
2. To fix the first two problems, and update some additional settings, in the *Manage* administrative menu, navigate to *Structure > Content types* (`admin/structure/types`). Then click *Manage display* in the dropdown button for the Vendor content type.

Content types		
Home » Administration » Structure		
+ Add content type		
NAME	DESCRIPTION	OPERATIONS
Basic page	Use basic pages for your static content, such as an 'About us' page.	Manage fields
Recipe	Recipe submitted by a vendor	Manage fields
Vendor	Information about a vendor	Manage fields Manage form display Manage display Edit Delete

3. Under the *Label* column, select *Hidden* for Main image. Do the same for Vendor URL.

Show row weights			
FIELD	LABEL	FORMAT	
⊕ Links		Visible ▾	
⊕ Body	- Hidden - ▾	Default ▾	
⊕ Vendor URL	- Hidden - ▾	Link ▾	Link text trimmed to 80 characters 
⊕ Main image	- Hidden - ▾	Image ▾	Original image 
Disabled			
No field is hidden.			

4. Click the gear for the Vendor URL field. Clear the *Trim link text length* setting. Links will no longer have a trim length. Check the *Open link in new window* checkbox. Now, when the link is clicked, it will open in a new browser window. Click *Update*.

Format settings: Link

Trim link text length

characters

Leave blank to allow unlimited link text lengths.

URL only

Show URL as plain text

Add rel="nofollow" to links

Open link in new window

Update **Cancel**

5. Drag the cross arrows next to the field items to reorder as Main image, *Body*, Vendor URL, and *Links*.

*You have unsaved changes.

FIELD	LABEL	FORMAT	
Main image	- Hidden -	Image	Original image
Body	- Hidden -	Default	
Vendor URL	- Hidden -	Link	Link text not trimmed Open link in new window
Links		Visible	
Disabled			
No field is hidden.			

6. Click **Save**.
7. Find the Vendor content item from step 1 again, and verify that the updates have been made.
8. Repeat similar steps to manage the display of the Recipe content type fields.

Expand your understanding

- Make the main image smaller. See [Bagian 6.13, "Setting Up an Image Style"](#).
- If you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Related concepts

[Bagian 6.12, "Concept: Image Styles"](#)

Additional resources

- [Drupal.org community documentation page "Specify how fields are displayed"](#)
- [Drupal.org community documentation page "Rearrange the order of fields"](#)
- [Drupal.org community documentation page "View modes"](#)

Attributions

Written by [Ann Greazel](#) and [Boris Doesborg](#).

6.12. Concept: Image Styles

Prerequisite knowledge

[Bagian 6.3, "Adding Basic Fields to a Content Type"](#)

What are image styles?

Image styles allow you to upload a single image but display it in several ways; each display variation, or *image style*, is the result of applying one or more *effects* to the original image.

As an example, you might upload a high-resolution image with a 4:3 aspect ratio, and display it scaled down, square cropped, or black-and-white (or any combination of these effects). The core software provides a way to do this efficiently:

1. Configure an image style with the desired effects on the *Image styles* page ([admin/config/media/image-styles](#)).
2. The effects will be applied the first time a particular image is requested in that style.
3. The resulting image is saved.
4. The next time that same style is requested, the saved image is retrieved without the need to recalculate the effects.

The core software provides several effects that you can use to define styles; others may be provided by contributed modules.

Visit the *Image styles* page via the *Manage* administrative menu, navigate to *Configuration > Media > Image styles* ([admin/config/media/image-styles](#)) to see the image styles that are defined by default.

Related topics

- [Bagian 6.13, "Setting Up an Image Style"](#)
- [Bagian 6.14, "Concept: Responsive Image Styles"](#)
- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)

Additional resources

[Drupal.org community documentation page "Working with images in Drupal 7 and 8"](#)

Attributions

Adapted and edited by [Boris Doesborg](#), and [Jojo Alphonso](#) at [Red Crackle](#), from ["Working with images in Drupal 7 and 8"](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#)

6.13. Setting Up an Image Style

Goal

Add an image style and use it to display images on Vendor pages.

Prerequisite knowledge

- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)
- [Bagian 6.11, "Changing Content Display"](#)
- [Bagian 6.12, "Concept: Image Styles"](#)

Site prerequisites

- Vendor and Recipe content types must exist. See [Bagian 6.1, "Adding a Content Type"](#).
- Main image fields must exist for both content types. See [Bagian 6.3, "Adding Basic Fields to a Content Type"](#).
- Content items must exist for both content types. See [Bagian 6.1, "Adding a Content Type"](#), [Bagian 6.3, "Adding Basic Fields to a Content Type"](#), and [Bagian 5.2, "Creating a Content Item"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Configuration > Media > Image styles* (`admin/config/media/image-styles`).
2. Click *Add image style*.
3. Enter the name *Extra medium (300x200)*
4. Click *Create new style*. The page *Edit style Extra medium (300x200)* appears.
5. In the *Effect* table, select *Scale and crop*. Click *Add*.
6. Fill in the fields as shown below.

Field name	Value
Width	300
Height	200

7. Click *Add effect*. The image style is saved with the chosen effects.

Edit style Extra medium (300x200) ☆

Home » Administration » Configuration » Media » Image styles

The image effect was successfully applied.

Preview

 original (view actual size)	 Extra medium (300x200) (view actual size)
800px	300px
600px	200px

Image style name*

Extra medium (300x200)

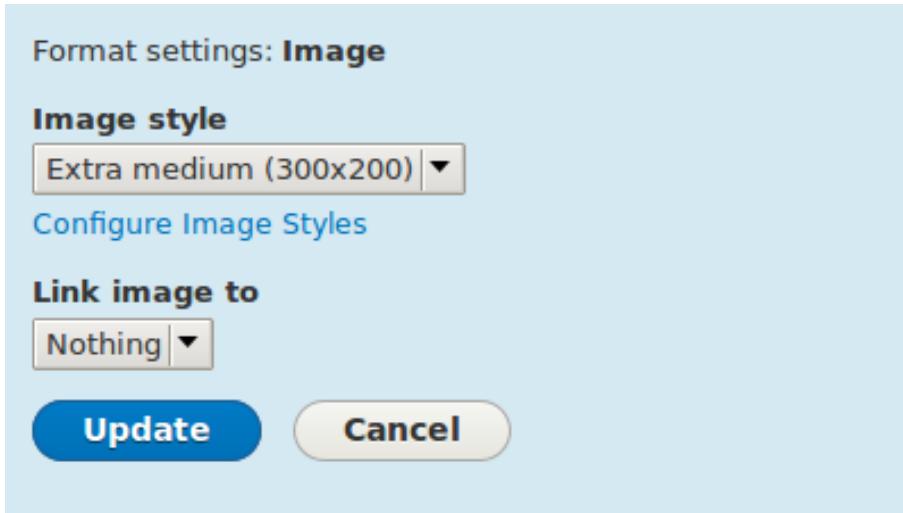
Machine name: extra_medium_300x200 [Edit]

Show row weights

EFFECT	OPERATIONS
+ Scale and crop 300×200	Edit ▼
+ Select a new effect ▾	Add
Update style Delete	

8. In the *Manage* administrative menu, navigate to *Structure > Content types* (`admin/structure/types`).
9. Click *Manage display* in the *Operations* dropdown for the Vendor content type. The *Manage display* page (`admin/structure/types/manage/vendor/display`) appears.
10. Ensure that the secondary tab *Default* is selected.
11. Click the cogwheel next to *Main image* for the configuration options.
12. Fill in the fields as shown below.

Field name	Value
Image style	Extra medium (300x200)
Link image to	Nothing



13. Click *Update*.
14. Click *Save*. The new image style will be used while displaying Vendor content.
15. Open a Vendor content item and verify that it now shows up with the scaled-down image. See [Bagian 5.3, "Editing a Content Item"](#) for information on how to locate an existing content item.
16. Repeat steps 8-15 for the Recipe content type.

Related concepts

- [Bagian 6.3, "Adding Basic Fields to a Content Type"](#)
- [Bagian 6.12, "Concept: Image Styles"](#)
- [Bagian 6.14, "Concept: Responsive Image Styles"](#)

Additional resources

[Drupal.org community documentation page "Working with images in Drupal 7 and 8"](#)

Attributions

Adapted and edited by [Boris Doesborg](#), and [Jojo Alphonso](#) at [Red Crackle](#) from ["Working with images in Drupal 7 and 8"](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

6.14. Concept: Responsive Image Styles

Prerequisite knowledge

- [Bagian 1.3, "Concept: Themes"](#)

- [Bagian 6.11, "Changing Content Display"](#)
- [Bagian 6.12, "Concept: Image Styles"](#)

What are responsive image styles?

The core Responsive Image module provides responsive image styles. This allows you to have images in your website that are specifically sized for different screen sizes. This is useful, for example, to make your site load faster on mobile devices because image sizes are optimized for smaller screens.

A responsive image style is a mapping between images styles and breakpoints. Breakpoints are the points where a responsive design needs to change in order to respond to different screen sizes. Responsive image styles can only be used if the breakpoints are defined in the theme.

When a responsive image style is defined, it can be used in the display settings for Image fields. This allows the site to display responsive images using the HTML5 picture tag.

Related topics

[Bagian 6.3, "Adding Basic Fields to a Content Type"](#)

Additional resources

- [Drupal.org community documentation page "Responsive web design"](#)
- [Drupal.org community documentation page "Responsive images in Drupal 8"](#)
- [Drupal.org community documentation page "Working with breakpoints in Drupal 8"](#)

Attributions

Adapted by [Boris Doesborg](#) from [Responsive images in Drupal 8](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

6.15. Concept: Text Formats and Editors

What are text formats and filters?

Text formats change how HTML tags and other text are processed and displayed on your site. Text formats are composed of a series of *filters*, each of which transforms text. When users create content, a text format is associated with the content, and the full, original text is stored in the database. The content is then passed through the filters in the text format before it becomes output on the site.

The core Filter module provides text format functionality, and the core Standard installation profile sets up *Basic HTML*, *Restricted HTML*, and *Full HTML* text formats. Each text format

has an associated permission, so that you can allow only trusted users to use permissive text formats. This restricts untrusted users to text formats like *Basic HTML*, which filters out dangerous HTML tags.

What are the editors associated with text formats?

Each text format can be associated with an editor, such as a visual WYSIWYG (What You See Is What You Get) HTML editor. The core Text Editor module provides the ability to associate editors with text formats, and to configure the editors (such as adding and removing buttons from their toolbars). The core CKEditor module provides the industry-standard editor known as CKEditor, so that it can be used to edit HTML content on your site.

What is cross-site scripting?

Cross-site scripting (XSS) is a security vulnerability typically found in websites. In a site that is not well protected, malicious users can enter script into web pages that are viewed by other users (for example, in a comment or in the body of a page). A cross-site scripting vulnerability may be used by attackers to login as another user. It is important to configure the text formats of your website to prevent such abuse.

Related topics

[Bagian 13.3, "Concept: Security and Regular Updates"](#)

Additional resources

- [Drupal.org community documentation page "Filter module: text formats that filter user input"](#)
- [Wikipedia page "Cross-site scripting"](#)

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

6.16. Configuring Text Formats and Editors

Goal

Add a horizontal rule tag to the *Basic HTML* text format, and a corresponding button to its editor configuration.

Prerequisite knowledge

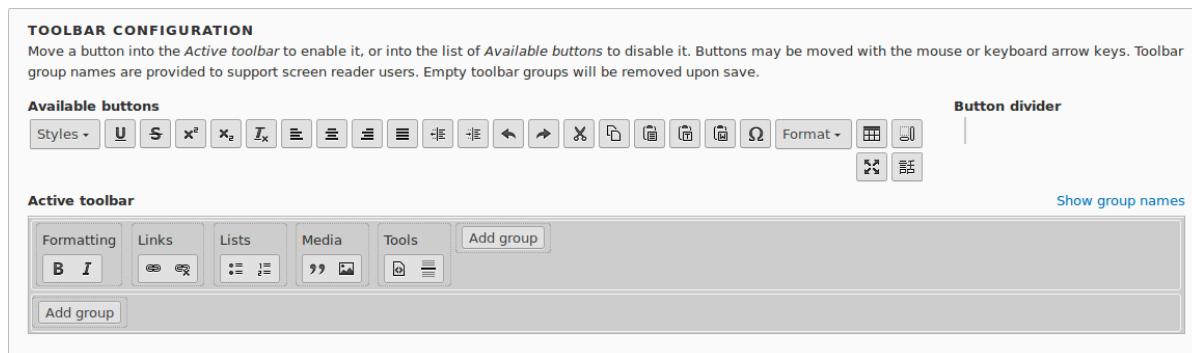
[Bagian 6.15, "Concept: Text Formats and Editors"](#)

Site prerequisites

- The core Filter, Editor, and CKEditor modules must be installed. They are installed on your site when you install with the core Standard installation profile.
- The *Basic HTML* text format must exist. This is created on your site when you install with the core Standard installation profile.

Steps

- In the *Manage* administrative menu, navigate to *Configuration > Content authoring > Text formats and editors* (`admin/config/content/formats`). The *Text formats and editors* page appears.
- Click *Configure* for the *Basic HTML* text format. The *Basic HTML* page appears.
- Note that *CKEditor* is selected in the *Text editor* field. This allows you to configure the editor's toolbar.
- Drag the *horizontal rule* button from *Available buttons* to *Tools* in *Active toolbar*. You may need to click the *Show group names* link to see the *Tools* group name.



- Note that you can change the *Filter processing order*.
- In the field *Allowed HTML tags*, add `<hr>` to what is already present.

Filter settings

Limit allowed HTML tags and correct faulty HTML

Enabled

Allowed HTML tags

<hr> <a href hreflang> <cite> <blockquote cite> <cc>

A list of HTML tags that can be used. By default only the *lang* and *dir* attributes are allowed for all HTML tags. Each HTML tag may have attributes which are treated as allowed attribute names for that HTML tag. Each attribute may allow all values, or only allow specific values. Attribute names or values may be written as a prefix and wildcard like *jump-**. JavaScript event attributes, JavaScript URLs, and CSS are always stripped.

Display basic HTML help in long filter tips

Add *rel="nofollow"* to all links

- Click *Save configuration*. You will be taken back to the *Text formats and editors* page. A message will be displayed saying that the text format has been updated.

 The text format *Basic HTML* has been updated.

Expand your understanding

If you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Attributions

Written by [Boris Doesborg](#).

Bab 7. Mengelola Akun Pengguna

7.1. Concept: Users, Roles, and Permissions

What are Users?

Anyone who visits your website is a *user*, including you. There are three groups of users:

- Users who are not logged in, or *anonymous users*
- Users who are logged in, or *authenticated users*
- The administrative user account that was automatically created when you installed your site, or User 1. See [Bagian 7.2, "Concept: The User 1 Account"](#).

What are Permissions?

The ability to do actions on your site (including viewing content, editing content, and changing configuration) is governed by *permissions*. Each permission has a name (such as *View published content*) and covers one action or a small subset of actions. A user must be granted a permission in order to do the corresponding action on the site; permissions are defined by the modules that provide the actions.

What are Roles?

Rather than assigning individual permissions directly to each user, permissions are grouped into *roles*. You can define one or more roles on your site, and then grant permissions to each role. The permissions granted to authenticated and anonymous users are contained in the *Authenticated user* and *Anonymous user* roles, and depending on the installation profile you used when you installed your site, there may also be an *Administrator* role that is automatically assigned all permissions on your site.

Each user account on your site is automatically given the *Authenticated user* role, and may optionally be assigned one or more additional roles. When you assign a role to a user account, the user will have all the permissions of the role when logged in.

It is a good practice to make several roles on your site. In the farmers market site example, you might want the following roles:

- A Vendor role that allows vendors to edit their own vendor listing page
- A Content editor role for editing the general farmers market pages
- A User manager role for managing the vendor accounts
- The *Administrator* role that was installed with your site, for expert users to manage the site configuration

Related topics

- [Bagian 7.3, "Creating a Role"](#)
- [Bagian 7.5, "Assigning Permissions to a Role"](#)
- [Bagian 7.6, "Changing a User's Roles"](#)
- [Bagian 7.4, "Creating a User Account"](#)
- [Bagian 7.2, "Concept: The User 1 Account"](#)
- [Bagian 7.7, "Assigning Authors to Content"](#)

Additional resources

- [Drupal.org community documentation page "Users, roles and permissions"](#)
- [Drupal.org community documentation page "Managing Users"](#)
- [Drupal.org community documentation page "User Roles"](#)

Attributions

Adapted by [Mark LaCroix](#), [Boris Doesborg](#), and [Jennifer Hodgdon](#) from "[User Roles](#)", copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

7.2. Concept: The User 1 Account

Prerequisite knowledge

[Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)

What is the user 1 account?

During the installation of your site, you created the first user account. Each user account internally has a numeric user ID, and since the ID of this user is one, it is commonly referred to as the *user 1* account. This user account is special, because independent of what roles it is assigned, someone logged in as user 1 has permission to do all actions on the site, including viewing and editing all content, editing any user account, changing site configuration, installing and uninstalling modules, and running the update script.

Because of this level of permission, some people refer to this account as the *root user*, similar to the "root" user account that has full administrative permissions in Linux and other operating systems.

It is usually better to make separate accounts for each administrative user, giving them the *Administrator* role, rather than having all administrative users log in using the user 1 account. There are several reasons for this:

- Some actions and updates on the site are logged, and if everyone uses the same account, it is difficult to know who did them if you have questions.
- The *Administrator* role permissions can be modified to be safer than the full permissions of the user 1 account, so that people do not inadvertently change site features that shouldn't be changed.
- People's responsibilities on a site may change over time. With ordinary user accounts, this can be mirrored in permissions by assigning or unassigning roles to their user accounts. If they are all using the user 1 account, this is more difficult.
- On some sites, the author of content or comments is displayed or tracked, and if everyone uses the same account to create content, it is difficult to know who created the content.

It is not possible to delete the user 1 account from the administrative user interface. It would be possible to do with a database query, but it could cause problems in your site and is not advisable.

Additional resources

[Drupal.org community documentation page "Accounts and roles"](#)

Attributions

Written and edited by [Mark LaCroix](#) and [Jennifer Hodgdon](#).

7.3. Creating a Role

Goal

Create a Vendor role to allow some - but not all - users to perform specific tasks.

Prerequisite knowledge

[Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)

Steps

1. In the *Manage* administrative menu, navigate to *People > Roles* (`admin/people/roles`).
2. You will find default roles *Anonymous user*, *Authenticated user*, and *Administrator* already present.

Roles ★

List Permissions Roles

Home » Administration » People

A role defines a group of users that have certain privileges. These privileges are defined on the [Permissions page](#). Here, you can define the names and the display sort order of the roles on your site. It is recommended to order roles from least permissive (for example, Anonymous user) to most permissive (for example, Administrator user). Users who are not logged in have the Anonymous user role. Users who are logged in have the Authenticated user role, plus any other roles granted to their user account.

+ Add role

Show row weights

NAME	OPERATIONS
⊕ Anonymous user	Edit ▾
⊕ Authenticated user	Edit ▾
⊕ Administrator	Edit ▾

Save

3. Click *Add Role* to add a custom role.
4. Type Vendor in the *Role name* field. Click *Save*.

Add role ★

Home » Administration » People » Roles

Role name*

Machine name: vendor [Edit]
The name for this role. Example: "Moderator", "Editorial board", "Site architect".

Save

5. You will see the message "Role Vendor has been added." displayed at the top of the page.

✓ Role Vendor has been added.

Expand your understanding

- [Bagian 7.5, "Assigning Permissions to a Role"](#)
- [Bagian 7.6, "Changing a User's Roles"](#)

Additional resources

[Drupal.org community documentation page "User Roles"](#)

Attributions

Adapted and edited by [Jack Probst](#), [Boris Doesborg](#), and [Joe Shindelar](#) from "[User Roles](#)", copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#)

7.4. Creating a User Account

Goal

Create Vendor user accounts for Sweet Honey and Happy Farm vendors.

Prerequisite knowledge

- [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)
- [Bagian 7.2, "Concept: The User 1 Account"](#)
- [Bagian 7.3, "Creating a Role"](#)

Site prerequisites

The Vendor role must exist on your site. See [Bagian 7.3, "Creating a Role"](#).

Steps

1. In the *Manage* administrative menu, navigate to *People* (`admin/people`).
2. Click *Add user*.

Add user ★

Home » Administration » People

This web page allows administrators to register new users. Users' email addresses and usernames must be unique.

Email address

A valid email address. All emails from the system will be sent to this address. The email address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by email.

Username *

Several special characters are allowed, including space, period (.), hyphen (-), apostrophe ('), underscore (_), and the @ sign.

Password *

Password strength:

Confirm password *

Passwords match:

Provide a password for the new account in both fields.

Status

- Blocked
 Active

Roles

- Authenticated user
 Administrator
 Vendor
 Notify user of new account

Picture

- Fill in the form fields. See the table below.

Field name	Explanation	Example value
Email address	A valid email address for the vendor. All emails from the system will be sent to this address. The email address is not made public.	honey@example.com
Username	A username for the vendor that they will use to sign in or author content items. Spaces are allowed; punctuation is not allowed except	Sweet Honey

Field name	Explanation	Example value
	for periods, hyphens, apostrophes, and underscores.	
Password	A password the vendor will use to sign in to the site. You can see how safe the password is on the <i>Password strength</i> gauge. You also get tips on how to make it safer.	(Make a secure password)
Confirm password	Type the same password to avoid any typing mistakes.	(Repeat password)
Status	Set the status of the user account. <i>Blocked</i> users will not be able to sign in.	Active
Roles	Set the role of the user account.	Vendor
Notify user of new account	Whether or not to send a notification to the vendor's email address.	Checked
Picture	Click <i>Browse</i> and select a picture to upload. Pay attention to size restrictions.	Photo of the vendor
Contact settings	Enable or disable the display of a contact form for the account.	Checked

- Click *Create new account*. You will get a notification about the user account creation.

✓ The image was resized to fit within the maximum allowed dimensions of 85x85 pixels.

A welcome message with further instructions has been emailed to the new user [Sweet Honey](#).

- Create a second Vendor account for Happy Farm by following the steps above.

Expand your understanding

Create a user account for yourself.

Attributions

Written by [Diána Lakatos](#) at [Pronovix](#).

7.5. Assigning Permissions to a Role

Goal

Change the permissions for the Vendor role so that users can create, edit, and delete Recipe and Vendor content, format the content, and contact each other.

Prerequisite knowledge

- [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)

Site prerequisites

The Vendor role must exist on your site. See [Bagian 7.3, "Creating a Role"](#).

Steps

1. In the *Manage* administrative menu, navigate to *People > Roles* (`admin/people/roles`). The *Roles* page appears.
2. Click *Edit permissions* in the dropdown for the Vendor role. The *Edit role* page appears where you can see all the available actions for the website such as, for example, *Post comments* or *Administer blocks*. The available permissions depend on the modules that are installed in the site. Note: Some permissions may have security implications. Be cautious while assigning permissions to roles.
3. Check the boxes for the following permissions, listed by module:

Module	Permission
Contact	Use users' personal contact forms
Filter	Use the Restricted HTML text format
Node	Recipe: Create new content
Node	Recipe: Edit own content
Node	Recipe: Delete own content
Node	Vendor: Edit own content
Quick Edit	Access in-place editing

PERMISSION	VENDOR
<i>Basic page:</i> View revisions	<input type="checkbox"/>
<i>Recipe:</i> Create new content	<input checked="" type="checkbox"/>
<i>Recipe:</i> Delete any content	<input type="checkbox"/>
<i>Recipe:</i> Delete own content	<input checked="" type="checkbox"/>
<i>Recipe:</i> Delete revisions Role requires permission to <i>view revisions</i> and <i>delete rights</i> for nodes in question, or <i>administer nodes</i> .	<input type="checkbox"/>
<i>Recipe:</i> Edit any content	<input type="checkbox"/>
<i>Recipe:</i> Edit own content	<input checked="" type="checkbox"/>
<i>Recipe:</i> Revert revisions Role requires permission <i>view revisions</i> and <i>edit rights</i> for nodes in question, or <i>administer nodes</i> .	<input type="checkbox"/>
<i>Recipe:</i> View revisions	<input type="checkbox"/>
<i>Vendor:</i> Create new content	<input type="checkbox"/>
<i>Vendor:</i> Delete any content	<input type="checkbox"/>
<i>Vendor:</i> Delete own content	<input type="checkbox"/>
<i>Vendor:</i> Delete revisions Role requires permission to <i>view revisions</i> and <i>delete rights</i> for nodes in question, or <i>administer nodes</i> .	<input type="checkbox"/>
<i>Vendor:</i> Edit any content	<input type="checkbox"/>
<i>Vendor:</i> Edit own content	<input checked="" type="checkbox"/>
<i>Vendor:</i> Revert revisions Role requires permission <i>view revisions</i> and <i>edit rights</i> for nodes in question, or <i>administer nodes</i> .	<input type="checkbox"/>
<i>Vendor:</i> View revisions	<input type="checkbox"/>
Path	

- Click *Save permissions*. You will get a message saying your changes have been saved.

✓ The changes have been saved.

Expand your understanding

- Log in as one of the new users you created in [Bagian 7.4, "Creating a User Account"](#). Verify whether you have the correct permissions.

- [Bagian 7.6, "Changing a User's Roles"](#)

Related concepts

[Bagian 7.2, "Concept: The User 1 Account"](#)

Additional resources

[Drupal.org community documentation page "Managing Users"](#)

Attributions

Adapted and edited by [Boris Doesborg](#), [Brian Emery](#), and [Jojo Alphonso](#) at [Red Crackle](#), from ["User Roles"](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

7.6. Changing a User's Roles

Goal

Change or add roles to a given user, either by editing a single-user or by applying a bulk operation.

Prerequisite knowledge

[Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)

Site prerequisites

The user account that you want to update, and the role you want it to have, must already exist. See [Bagian 7.4, "Creating a User Account"](#), [Bagian 7.3, "Creating a Role"](#), and [Bagian 7.5, "Assigning Permissions to a Role"](#).

Steps

Updating the roles using single-user editing method

1. In the *Manage* administrative menu, navigate to *People (admin/people)*.
2. Locate the user 1 account (named "admin") to assign it the *Administrator* role. If it is not immediately visible, use the *Name or email contains* filter, or other filters, to narrow down the list.
3. Click *Edit* to update the user account.

The screenshot shows the 'People' page with the following details:

USERNAME	STATUS	ROLES	MEMBER FOR	LAST ACCESS	OPERATIONS
Happy Farm	Active	• Vendor	5 seconds	never	<button>Edit</button>
Sweet Honey	Active	• Vendor	8 seconds	never	<button>Edit</button>
admin	Active	• Administrator	11 minutes 6 seconds	2 minutes 37 seconds ago	<button>Edit</button>

- On the *Edit* page, scroll down to *Roles* section. Check the *Administrator* role for the user account.

Roles

- [Authenticated user](#)
- [Administrator](#)
- [Vendor](#)

- Click *Save* to update the user account. You should be returned to the *People* page and see a message saying that the changes have been saved.

The changes have been saved.

Updating the roles using bulk editing method

- If the users Happy Farm and Sweet Honey did not already have the Vendor role, here is how you would add it. In the *Manage* administrative menu, navigate to *People* (*admin/people*).
- Locate Vendor user accounts *Sweet Honey* and *Happy Farm* and check them. If they are not immediately visible, use the *Name or email contains* filter, or other filters, to narrow down the list.
- Select *Add the Vendor role to the selected users* from the *Action* select list.

Action
Add the Vendor role to the selected users ▾

Apply to selected items

<input type="checkbox"/>	USERNAME	STATUS	ROLES	MEMBER FOR	LAST ACCESS	OPERATIONS
<input checked="" type="checkbox"/>	Happy Farm	Active	• Vendor	12 seconds	never	<button>Edit</button>
<input checked="" type="checkbox"/>	Sweet Honey	Active	• Vendor	14 seconds	never	<button>Edit</button>
<input type="checkbox"/>	admin	Active	• Administrator	3 days 2 hours	54 seconds ago	<button>Edit</button>

Apply to selected items

✓ Add the Vendor role to the selected users was applied to 2 items.

4. Click *Apply to selected items*. You should see a message indicating that the desired changes were made.

Attributions

Written by [Chris Dart](#) and [Jennifer Hodgdon](#)

7.7. Assigning Authors to Content

Goal

Assign Vendor content items Happy Farm and Sweet Honey to the corresponding Vendor user accounts, so they can edit their own Vendor profiles on the site.

Prerequisite knowledge

- [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#)

Site prerequisites

- The Vendor content type must exist, and your site must have at least two Vendor content items. See [Bagian 6.1, "Adding a Content Type"](#), [Bagian 6.3, "Adding Basic Fields to a Content Type"](#), and [Bagian 5.2, "Creating a Content Item"](#).
- User accounts for at least two vendors must exist. See [Bagian 7.4, "Creating a User Account"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Content* (`admin/content`).
2. Find Vendor content item Happy Farm in the list. If it is not immediately visible, you can filter the list by *Published status*, *Content type* (Vendor), *Title*, or *Language*. Click *Edit* for the Vendor content item you would like to assign an author to.

3. Locate the information block and click *Authoring information*.

The screenshot shows the 'Authoring information' section of a publishing interface. It includes fields for last save date and time, author, revision log message, URL path settings, and promotion options.

Published
Last saved: 08/08/2016 - 14:02
Author: Happy Farm
 Create new revision

Revision log message
Briefly describe the changes you have made.

URL PATH SETTINGS

AUTHORING INFORMATION

Authored by
Happy Farm (3)

The username of the content author.

Authored on
2016-08-08 14:00:52

Format: 2016-08-08 14:02:57. Leave blank to use the time of form submission.

PROMOTION OPTIONS

4. Start typing the Vendor's user name Happy Farm in the *Authored by* field. The field lists matching user names. Select the Vendor's user name from the list.
5. Click *Save and keep published*.
6. You will get a notification that the Vendor content item has been updated.

Vendor Happy Farm has been updated.

7. Follow these steps again to assign Vendor content item Sweet Honey to the Vendor user account Sweet Honey.

Attributions

Written by [Diána Lakatos](#) at [Pronovix](#).

Bab 8. Blok

8.1. Concept: Blocks

What is a block?

Blocks are individual pieces of your site's web page layout. They are placed inside the regions (see [Bagian 2.1, "Concept: Regions in a Theme"](#)) of your theme, and can be created, removed, and rearranged in the *Block layout* (`admin/structure/block`) administration page. Examples of blocks include the *Who's online* listing, the main navigation menu, and the breadcrumb trail. The main page content is also a block.

Some modules make new blocks available for placement on your site. For example, when the core Search module is installed and configured, it provides a block that contains a search form. You may also create and place your own custom blocks.

Each block has its own configuration settings, which allow you to select which pages of your site will display the block. It is even possible to place multiple copies of a block, each with its own separate configuration and visibility rules.

Related topics

- [Bagian 2.1, "Concept: Regions in a Theme"](#)
- [Bagian 8.2, "Creating A Custom Block"](#)
- [Bagian 8.3, "Placing a Block in a Region"](#)

Attributions

Adapted by [Les Lim](#) from ["Working with blocks \(content in regions\)"](#) copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

8.2. Creating A Custom Block

Goal

Create a block showing the hours and location of the farmers market.

Prerequisite knowledge

- [Bagian 8.1, "Concept: Blocks"](#)
- [Bagian 2.1, "Concept: Regions in a Theme"](#)

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Block layout > Custom block library* (`admin/structure/block/block-content`).
2. Click *Add custom block*. The *Add custom block* page appears.
3. Fill in the fields as shown below.

Field name	Value
Block description	Hours and location block
Body	Open: Sundays, 9 AM to 2 PM, April to September Location: Parking lot of Trust Bank, 1st & Union, downtown Anytown.

Add custom block ☆

[Home](#)

Block description *

Hours and location block

A brief description of your block.

Body

Text format Basic HTML ▾ [About text formats](#) ?

Revision information

New revision Create new revision

Revision log message

Briefly describe the changes you have made.

Save

4. Click *Save*. A message appears indicating the block has been saved.

Expand your understanding

- Edit the content of your custom block. In the *Manage* administrative menu, navigate to *Structure > Block layout > Custom block library* (`admin/structure/block/block-content`). Find your block in the list and click *Edit* to make changes.
- Place the block you created in the sidebar. See [Bagian 8.3, "Placing a Block in a Region"](#) for details.

Additional resources

[Drupal.org community documentation page "Working with blocks \(content in regions\)"](#)

Attributions

Adapted by [Jacob Redding](#) and [Boris Doesborg](#) from [Working with blocks \(content in regions\)](#), copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

8.3. Placing a Block in a Region

Goal

Place the Opening hours and location block in the website's sidebar.

Prerequisite knowledge

[Bagian 8.1, "Concept: Blocks"](#)

Site prerequisites

- The core Bartik theme must be installed and set as default. See [Bagian 4.6, "Configuring the Theme"](#).
- The Opening hours and location block must exist. See [Bagian 8.2, "Creating A Custom Block"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Block layout* (`admin/structure/block`). The *Block layout* page appears, listing the theme's regions.
2. Ensure that in the secondary tab the core Bartik theme is selected. Block placement is defined per theme.

3. Locate the region *Sidebar second* in the list and click *Place block* next to it. The *Place block* window appears, listing all the blocks.
4. Locate the block *Opening hours and location* and click *Place block* next to it. The *Configure block* window appears. Fill in the fields as shown below.

Field name	Value
Title	Hours and location
Display title	Checked
Region	Sidebar second

You can also hide or display the block on specific pages. In the case of the Farmer's market website you do not set any of these configuration options because you want to show the block everywhere.

The screenshot shows the 'Configure block' page for a 'Basic block Hours and location block'. The page includes a success message, block description, title, display title, visibility settings for content types and roles, region selection, and a save button.

Configure block ☆

Home » Administration » Structure » Block layout » Configure block

✓ Basic block *Hours and location* block has been created.

Block description: Hours and location block

Title *
Hours and location Machine name: hours_location [Edit]

Display title

Visibility

Content types Not restricted	Content types <input type="checkbox"/> Basic page <input type="checkbox"/> Recipe <input type="checkbox"/> Vendor
Pages Not restricted	
Roles Not restricted	

Region
Sidebar second ▾

Select the region where this block should be displayed.

Save block

5. Click *Save block*. The *Block layout* page appears. You can drag blocks to change the order in which they will appear.
6. Verify that the *Opening hours and location* block is listed in the *Sidebar second* region, and click *Save blocks*.

The block has been placed on the sidebar of all pages that use the core Bartik theme.

My account Log out

ANYTOWN FARMERS MARKET

Farm Fresh Food

Home About

Home » Node

Tools

Add content

About

View Edit Delete Revisions

City Market started in April 1990 with five vendors.

Today, it has 100 vendors and an average of 2000 visitors per day.

Hours and location

Open: Sundays, 9 AM to 2 PM, April to September

Location: Parking lot of Trust Bank, 1st & Union, downtown

Expand your understanding

- Remove the *Powered by Drupal* block from the *Footer fifth* region by setting the region to - *None* -.
- Remove the *Tools* block from the *Sidebar first* region by setting the region to - *None* -. This block is only visible to logged-in users.
- Place the *User login* block in a region.
- If you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

Bab 9. Membuat Daftar dengan Views

9.1. Concept: Uses of Views

Prerequisite knowledge

- [Bagian 2.4, "Concept: Modular Content"](#)
- [Bagian 2.3, "Concept: Content Entities and Fields"](#)

What is a view?

A *view* is a listing of content on a website. The core Views module handles the display of views, and the core Views UI module allows you to create and edit them in the administrative interface. When you define views, you are interested in taking data from your website and displaying it to the user.

What types of data can be displayed using views?

You can create views to output practically any content entity that is stored in the system. For example, you can create the following lists for the farmers market site:

- Lists of vendors
- Lists of recipes
- Lists of the most recent content on the site
- Lists of users on the site

What are the ways data can be output using views?

A listing created by a view can be in any of the following forms:

- Table with sortable fields
- Grid layouts
- Teasers or pictures that link to articles
- Blocks
- JSON output
- RSS feeds
- Calendars
- On-screen slideshows

Related topics

- [Bagian 9.2, "Concept: The Parts of a View"](#)

- [Bagian 9.3, "Creating a Content List View"](#)

Attributions

Written and edited by [Michael Lenahan](#) at [erdfisch](#), and [Jojo Alphonso](#) at [Red Crackle](#).

9.2. Concept: The Parts of a View

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 2.4, "Concept: Modular Content"](#)
- [Bagian 9.1, "Concept: Uses of Views"](#)
- [Bagian 5.1, "Concept: Paths, Aliases, and URLs"](#)
- [Bagian 8.1, "Concept: Blocks"](#)

What are the parts of a view?

When you are editing a view in the administrative interface, you will see the following parts (or sections), which allow you to specify what data to output, in what order, and in what format:

Display

Each view can have one or more displays, each of which produces one type of output. Options for display types include:

Page

Makes output at a particular URL, for the main page content at that URL.

Block

Makes output in a block, which can be placed on pages.

Feed

Makes an RSS or another type of feed.

Attachment

Makes output that you can attach to another display.

Format

Depending on the display type, you can choose to output your data in a table, grid, HTML list, or another format. Some formats also give you a second choice that lets you output either rendered entities or fields; other formats do not give you this choice (for example, if you use a table format, you must always use fields).

Fields

Depending on the format choice, you may be able to choose which content fields are output. For example, if you were making a view of recipe content items, in a block display you might show only the recipe names, while in a full page display you might also show an image field because you have more space.

Filter criteria

Filters limit the data to be output, based on criteria such as whether the content has been published or not, the type of content, or a field value. For instance, to make a view of recipe content items, you would need to filter to the Recipe content type, and to published recipes. Filters can also be *exposed*, which means that users will have a form where they can choose their own filter values. You might use this on a Recipe page to let users filter for recipes with certain ingredients.

Sort criteria

Defines the order to present the output, which can be based on any content field.

Contextual filters

Contextual filters are like regular filters, except that the values come from the *context* of the view display, such as the full URL of the page being displayed, the current date or time, or some other value that can be detected by the view calculation.

Relationships

Relationships allow you to expand what is displayed in your view, by relating the base content being displayed to other content entities. Relationships are created using fields on the base content that relate it to the other content; one example is that all regular content items have an author field, which references the user account of the person who authored the content. Once you have created a relationship, you can display fields from the referenced entity in the view.

Related topics

[Bagian 9.3, "Creating a Content List View"](#)

Attributions

Written and edited by [Surendra Mohan](#) and [Jennifer Hodgdon](#).

9.3. Creating a Content List View

Goal

Create a page listing vendors that will be automatically updated whenever a vendor is added, deleted, or updated on the site.

Prerequisite knowledge

- [Bagian 9.1, "Concept: Uses of Views"](#)
- [Bagian 9.2, "Concept: The Parts of a View"](#)

Site prerequisites

- The core Views and Views UI modules must be installed. These are installed for you when you install with the core Standard installation profile.
- The Vendor content type must exist, with URL and Main image fields. Your site must have a couple of Vendor content items. See [Bagian 6.1, "Adding a Content Type"](#), [Bagian 6.3, "Adding Basic Fields to a Content Type"](#), and [Bagian 5.2, "Creating a Content Item"](#).
- The *Medium (220x220)* image style must be defined. This is created on your site when you install the core Image module (installed with the core Standard installation profile) but can be recreated if deleted. See [Bagian 6.13, "Setting Up an Image Style"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Views > Add view* (*admin/structure/views/add*). The *Add view* wizard appears.
2. Fill in the fields as shown below.

Field name	Explanation	Example value
View name	Name of the view that will be visible in the administration pages	Vendors
Show	Type of information listed in the view	Content
of type	Specify content type	Vendor
sorted by	List order	Title
Create a page	Create a page that displays the view	Checked
Page title	Title show above the view	Vendors
Path	Address of the page	vendors
Display format	Type of list	Table
Items to display	Number of items visible on the page	10
Use a pager	Split up the list in several pages if there are more items	Checked
Create a menu link	Add the view page to the menu	Checked

Field name	Explanation	Example value
Menu	Menu in which to add the link	Main navigation
Link text	Label of the link in the menu	Vendors

The screenshot shows the 'Add view' configuration page. It includes sections for basic information, view settings, page settings, and page display settings. The 'View basic information' section has a 'View name' field set to 'Vendors'. The 'View settings' section shows 'Content' as the type and 'Vendor' as the subtype, with 'sorted by' set to 'Title'. The 'Page settings' section has 'Create a page' checked, 'Page title' set to 'Vendors', and 'Path' set to 'vendors'. The 'Page display settings' section has 'Display format' set to 'Table' and 'of' set to 'teasers', with 'Items to display' set to 10.

- Click *Save and edit*. The view configuration page appears.
- Click *Add* from the dropdown button in the *Fields* section. The *Add fields* pop-up appears.
- Enter the word "image" in the search field.
- Check Main image in the table.
- Click *Apply*. The *Configure field: Content: Main Image* pop-up appears.
- Fill in the fields as shown below.

Field name	Explanation	Example value
Create a label	Add a label before the field value	Unchecked
Image style	The format of the image	Medium (220x220)
Link image to	Add a link to the content item	Content

- Click *Apply*. The view configuration page appears.
- Click *Add* from the dropdown button in the *Fields* section. The *Add fields* pop-up appears.

11. Enter the word "body" in the search field.
12. Select *Body* in the table.
13. Click *Apply*. The *Configure field: Content: Body* pop-up appears.
14. Fill in the fields as shown below.

Field name	Explanation	Example value
Create a label	Add a label before the field value	Unchecked
Formatter	The presentation of the field value	Summary or trimmed
Trimmed limit:	The number of maximum characters shown	120

15. Click *Apply*. The view configuration page appears.
16. Click *Content: Title (Title)* in the *Fields* list. The *Configure field: Content: Title* pop-up appears.
17. Uncheck *Create a label*. This will remove the label that was created by the wizard.
18. Click *Apply*. The view configuration page appears.
19. Click *Rearrange* from the dropdown button in the *Fields* section. The *Rearrange fields* pop-up appears.
20. Drag the fields into the right order: Image, Title, Body.
21. Click *Apply*. The view configuration page appears.
22. Optionally, click *Update preview* for a preview.
23. Click *Save*.

Membuat Daftar dengan Views

Vendors (Content) ☆

Home » Administration » Structure » Views » Vendors » Edit

The view *Vendors* has been saved.

Displays

Page + Add Edit view name/description

Display name: Page View Page

TITLE
Title: Vendors

FORMAT
Format: Table | Settings

FIELDS Add
Content: Main image
Content: Title
Content: Body

FILTER CRITERIA Add
Content: Publishing status (= Yes)
Content: Content type (= Vendor)

SORT CRITERIA Add
Content: Title (asc)

PAGE SETTINGS
Path: /vendors
Menu: Normal: Vendors
Access: Permission | View published content

HEADER Add

FOOTER Add

NO RESULTS BEHAVIOR Add

PAGER
Use pager: Mini | Mini pager, 10 items
More link: No

Save Cancel

24. Navigate to the homepage and click Vendors from the main navigation to see the result.

Anytown Farmers Market
Farm Fresh Food

My account Log out

Home About Vendors

Home

Tools Add content

Vendors

Image	Vendor Name	Description
	Happy Farm	Happy Farm grows vegetables that you will love.
	Sweet Honey	Sweet Honey produces honey in a variety of flavors throughout the year.

Hours and location

Open: Sundays, 9 AM to 2 PM, April to September

Location: Parking lot of Trust Bank, 1st & Union, downtown

Expand your understanding

The link to the view in the main navigation will probably not be in the right place. Change the order of the menu items in the main navigation. See [Bagian 5.8, "Changing the Order of Navigation"](#).

Attributions

Written/edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

9.4. Duplicating a View

Goal

Create a page listing recipes by duplicating the existing Vendors view. Modify the page so that the recipes are displayed in a grid and can be filtered by ingredients.

Prerequisite knowledge

- [Bagian 9.1, "Concept: Uses of Views"](#)
- [Bagian 9.2, "Concept: The Parts of a View"](#)
- [Bagian 9.3, "Creating a Content List View"](#)

Site prerequisites

- The Vendor and Recipe content types must exist; both must have Main image fields, and the Recipe content type must have an Ingredients field. Your site must also have a couple of Recipe content items. See [Bagian 6.1, "Adding a Content Type"](#), [Bagian 6.3, "Adding Basic Fields to a Content Type"](#), [Bagian 6.6, "Setting Up a Taxonomy"](#), [Bagian 6.9, "Changing Content Entry Forms"](#), and [Bagian 5.2, "Creating a Content Item"](#).
- The Vendors view must exist. See [Bagian 9.3, "Creating a Content List View"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Views* (`admin/structure/views`). Find the view "Vendors" and click *Duplicate* in its dropdown button.

Membuat Daftar dengan Views

The screenshot shows the 'Views' configuration page. At the top, there are 'List' and 'Settings' tabs. Below them is a breadcrumb navigation: Home > Administration > Structure. A blue button '+ Add view' is visible. A search bar 'Filter by view name or description' is present. The main area is titled 'Enabled' and contains a table with four columns: VIEW NAME, DESCRIPTION, TAG, and PATH. The first row is 'Taxonomy term', which displays 'Feed, Page' and has a machine name 'taxonomy_term'. The description says 'Content belonging to a certain taxonomy term.' The tag is 'default' and the path is '/taxonomy/term/%/feed, /taxonomy/term/'. An 'Edit' button with a dropdown arrow is shown. The second row is 'Vendors', displaying 'Page' and having a machine name 'vendors'. The description is 'Displays: Page'. The tag is 'vendors' and the path is '/vendors'. An 'Edit' button with a dropdown arrow is shown. The third row is 'Who's new', displaying 'Block' and having a machine name 'who_s_new'. The description says 'Shows a list of the newest user accounts on the site.' The tag is 'default' and the path is '/'. An 'Edit' button with a dropdown arrow is shown. The fourth row is 'Who's online block', displaying 'Block' and having a machine name 'who_s_online'. The description says 'Shows the user names of the most recently active users, and the total number of active users.' The tag is 'default' and the path is '/'. An 'Edit' button with a dropdown arrow is shown. To the right of the table, a vertical context menu is open, listing 'Edit', 'Duplicate', 'Disable', and 'Delete'.

VIEW NAME	DESCRIPTION	TAG	PATH	OPERATIONS
Taxonomy term Displays: Feed, Page Machine name: taxonomy_term	Content belonging to a certain taxonomy term.	default	/taxonomy/term/%/feed, /taxonomy/term/%	Edit ▾
Vendors Displays: Page Machine name: vendors		vendors		Edit ▾
Who's new Displays: Block Machine name: who_s_new	Shows a list of the newest user accounts on the site.	default		Edit ▾
Who's online block Displays: Block Machine name: who_s_online	Shows the user names of the most recently active users, and the total number of active users.	default		Edit ▾

2. Name the duplicate "Recipes" and click *Duplicate*. The view configuration page appears.
3. Change the title of the view page to "Recipes" by clicking Vendors in the *Title* section. The *Page: The title of this view* pop-up appears. Type in "Recipes". Click *Apply*.

The screenshot shows the 'Title' section of the view configuration page. The title field contains 'Recipes'. Below it, a note says 'This title will be displayed with the view, wherever titles are normally displayed; i.e., as the page title, block title, etc.' There are 'Apply' and 'Cancel' buttons at the bottom.

Title
Recipes
This title will be displayed with the view, wherever titles are normally displayed; i.e., as the page title, block title, etc.
Apply **Cancel**

4. Change from a table to a grid format by clicking *Table* in the *Format* section. The *Page: How should this view be styled* pop-up appears. Check *Grid* and click *Apply*. The *Page: Style options* pop-up appears. Retain the default values and click *Apply*.
5. Retain only the title and image fields for the Recipes view by clicking *Content: Body* in the *Fields* section. Click *Remove* in the pop-up that appears.
6. Change the content type filter to use the Recipe content type by clicking *Content: Type (=Vendor)* in the *Filter criteria* section. In the *Configure filter criterion: Content: Type* pop-up, check Recipe and uncheck Vendor. Click *Apply*.
7. Add a further filter that is exposed to visitors by clicking *Add* in the *Filter criteria* section. Search for "ingredients" and check "Ingredients (field_ingredients)". Click *Add and configure filter criteria*.
8. The appearing pop-up offers extra settings on vocabulary and selection type. Click *Apply and continue*. The next pop-up allows you to expose this filter to visitors. Fill in the fields as shown below, and click *Apply*.

Field name	Explanation	Example value
Expose this filter to visitors, to allow them to change it	Allow visitors to filter and search	Checked
Required	Whether a value has to be provided or not	Unchecked
Label	Label shown for this filter on the view page	Find recipes using...

Appears in: recipe.

Expose this filter to visitors, to allow them to change it

Filter type to expose

- Single filter
- Grouped filters

Grouped filters allow a choice between predefined operator|value pairs.

Required

Label

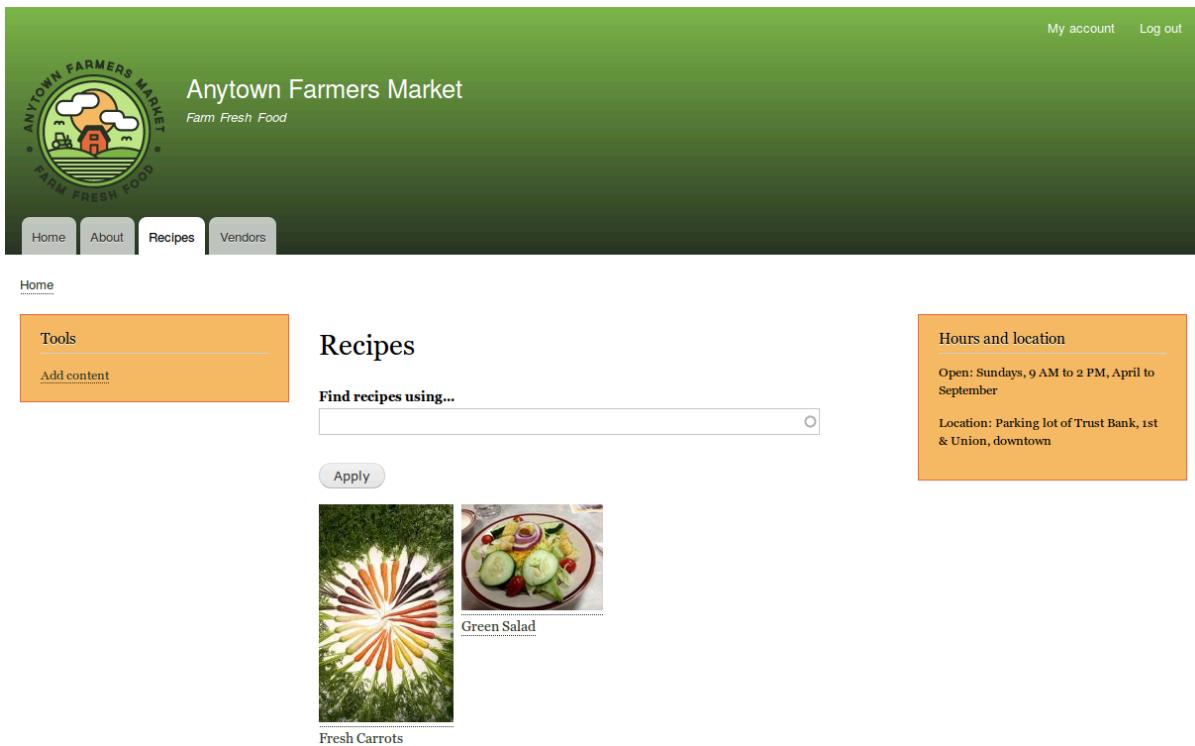
Find recipes using...

Description

9. Change the *Path* label field to "Recipes" by clicking "/vendors" in the *Page settings* section. In the pop-up that appears, enter the path "recipes" and click *Apply*.

Note that when editing a view, you enter paths without the leading "/", unlike on other administrative pages (such as when providing a path to a content item page).

10. Change the menu link title by clicking "Normal: Vendors" in the *Page settings* section. In the pop-up that appears, change the title to "Recipes" and click *Apply*.
11. In order to use Ajax (see [Ajax entry in the Glossary](#)) to make filtering and paging faster for users, click *Advanced* to expand the section for more options. In the *Other* section, change *Use AJAX: No* to "Yes" by clicking *No* and checking *Use AJAX* in the pop-up that appears. Click *Apply*.
12. Click *Save* to save the view.
13. Go back to the home page and click *Recipes* in the navigation to view the new Recipes page.



Expand your understanding

The link to the view in the main navigation will probably not be in the right place. Change the order of the menu items in the main navigation. See [Bagian 5.8, "Changing the Order of Navigation"](#).

Related concepts

- [Bagian 2.5, "Planning your Content Structure"](#)
- [Ajax entry in the Glossary](#)

Attributions

Written and edited by [Laura Vass](#) at [Pronovix](#), and [Jojo Alphonso](#) at [Red Crackle](#).

9.5. Adding a Block Display to a View

Goal

Add a block display to the Recipes view to display the most recent recipes in a sidebar, and change its configuration without changing the existing Recipes page view.

Prerequisite knowledge

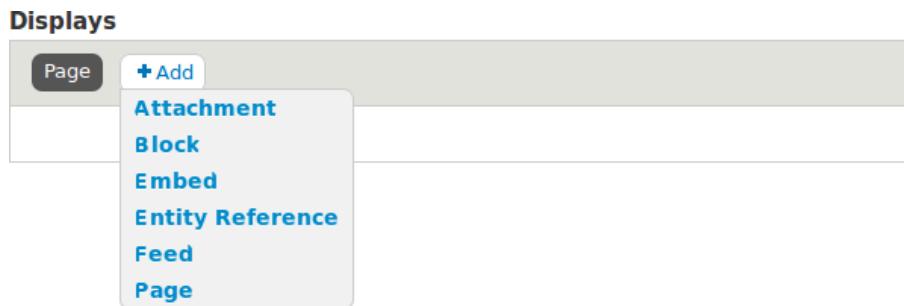
- [Bagian 9.1, "Concept: Uses of Views"](#)
- [Bagian 9.2, "Concept: The Parts of a View"](#)
- [Bagian 9.3, "Creating a Content List View"](#)

Site prerequisites

- The Recipe content type must exist, it must have a Main image field, and your site must have a couple of Recipe content items. See [Bagian 6.1, "Adding a Content Type"](#), [Bagian 6.3, "Adding Basic Fields to a Content Type"](#), [Bagian 6.9, "Changing Content Entry Forms"](#), and [Bagian 5.2, "Creating a Content Item"](#).
- The *Thumbnail (100x100)* image style must be defined. This is created on your site when you install the core Image module (installed with the core Standard installation profile) but can be recreated if deleted. See [Bagian 6.13, "Setting Up an Image Style"](#).
- The Recipes view must exist. See [Bagian 9.3, "Creating a Content List View"](#) and [Bagian 9.4, "Duplicating a View"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Structure > Views (admin/structure/views)*. Find the view "Recipes" and click *Edit* from its dropdown button. Alternatively, navigate to the Recipes page in the main site navigation, and click the *Edit view* contextual link in the main area of the page. See [Bagian 4.1, "Concept: Administrative Overview"](#) for information about contextual links.
2. Create a new block display by clicking *Add* in the *Displays* section. Click *Block* from the dropdown button. The new display is created, and the focus is automatically switched to its configuration.



3. Change the title of this display by clicking *Block* next to *Display name*. The *Block: The name and the description of this display* pop-up appears. Change the *Administrative name* to "Recent recipes". Click *Apply*.
4. Change the title of the view by clicking *Title: Recipes* in the *Title* section. In the pop-up that appears, select *This block (override)* from the *For* select list. Change the title to "New recipes" and click *Apply (this display)*.



5. Change the view's style by clicking *Format: Grid* in the *Format* section. In the pop-up that appears, select *This block (override)* from the *For* select list. Select *Unformatted list* and Click *Apply (this display)*. You can further configure the style options in the next pop-up that appears. Then click *Apply*.
6. Configure the image field by clicking *Content: Main image* in the *Fields* section. In the pop-up that appears, select *This block (override)* from the *For* select list. Select *Image style Thumbnail (100x100)*. Click *Apply (this display)*.

Configure field: Content: Main image

★

Home » Administration » Structure » Views

For

This block (override) ▾

Appears in: vendor, recipe.

Create a label

Label

Place a colon after the label

Exclude from display

Enable to load this field as hidden. Often used to group fields, or to use as token in another field.

Column used for click sorting

target_id ▾

Used by Style: Table to determine the actual column to click sort the field on. The default is usually fine.

Formatter

Image ▾

Image style

Thumbnail (100×100) ▾

[Configure Image Styles](#)

Link image to

Content ▾

STYLE SETTINGS

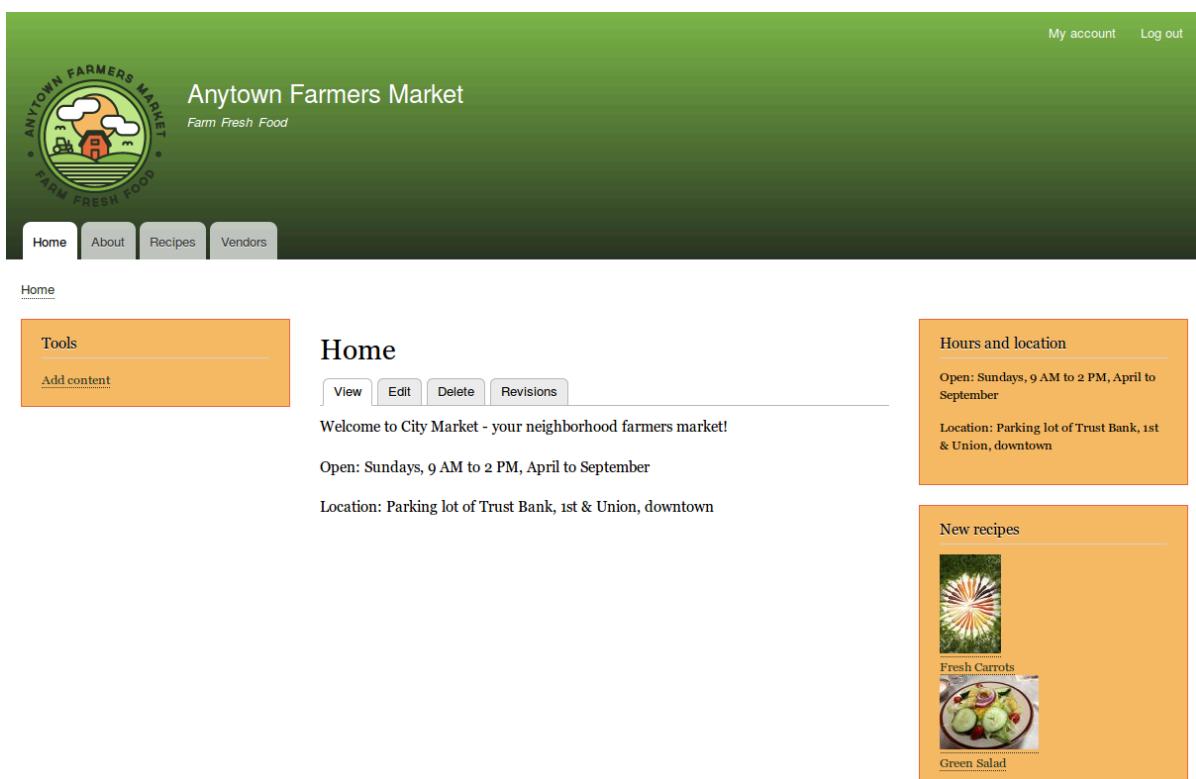
REWRITE RESULTS

-
7. Remove ingredients as a filter by clicking *Content: Ingredients (exposed)* in the *Filter criteria* section. In the pop-up that appears, select *This block (override)* from the *For* select list. Click *Remove* at the bottom.

8. Configure how you want the content to be sorted in the view by clicking *Add* in the *Sort criteria* section. In the pop-up that appears, select *This block (override)* from the *For* select list. Check the field *Authored on* (from Content category), and then *Add and configure sort criteria*.
9. In the appearing configuration pop-up, select *Sort descending* to have the most recent recipes appear first. Click *Apply*.
10. Specify the number of items to be displayed by clicking *Use pager: Mini* in the *Pager* section. In the pop-up that appears, select *This block (override)* from the *For* select list. Under *Pager*, select *Display a specified number of items*. Click *Apply (this display)*. In the *Block: Pager options* pop-up, provide "5" as the value for *Items to display*. Click *Apply*.
11. Click *Save*. You will either see the view editing page again, or the Recipes page, depending on what you did in step 1. You should also see a message saying that the view has been saved.

 The view *Recipes* has been saved.

12. Place the "Recipes: Recent Recipes" block in the *Sidebar second* region. See [Bagian 8.3, "Placing a Block in a Region"](#). Navigate to the site's home page to see the block.



The screenshot shows the homepage of the Anytown Farmers Market website. At the top, there is a navigation bar with links for Home, About, Recipes, and Vendors. Below the navigation is a main content area with a logo for "Anytown Farmers Market" featuring a sun, clouds, and a farm building, and the text "Farm Fresh Food". To the right of the logo is the page title "Anytown Farmers Market" and the subtitle "Farm Fresh Food". Below the title, there is a "Tools" menu with "Add content" and a "Home" link. The main content area contains a welcome message: "Welcome to City Market - your neighborhood farmers market!", opening hours ("Open: Sundays, 9 AM to 2 PM, April to September"), and location information ("Location: Parking lot of Trust Bank, 1st & Union, downtown"). To the right of the main content are two sidebar blocks: "Hours and location" (listing opening hours and location) and "New recipes" (showing thumbnail images for "Fresh Carrots" and "Green Salad").

Attributions

Written and edited by [Laura Vass](#) at Pronovix, [Jennifer Hodgdon](#), and [Joy Alphonso](#) at Red Crackle.

Bab 10. Membuat Situs Anda Multilingual

10.1. Concept: User Interface, Configuration, and Content translation

Prerequisite knowledge

- [Bagian 1.5, "Concept: Types of Data"](#)
- [Bagian 1.2, "Concept: Modules"](#)
- [Bagian 2.3, "Concept: Content Entities and Fields"](#)

What can be translated on your site?

The base language for the software that your site runs (core software, modules, and theme) is English. However, you can create either a site that displays in a different language or a multi-lingual site; if you do so, all relevant text, images, and file attachments should be shown in the appropriate language for each site visitor. You need to have the core Language module installed in order to use a language other than English on the site.

There are three types of information that you can translate, each with its own method for translating:

User interface text

Built-in text present in the core software, modules, and themes. This can be translated from the base English language of the software into the language(s) of your site. Typically, rather than needing to translate this text yourself, you can download translations. You need to have the core Interface Translation module installed in order to translate this text.

Configuration text

Text whose structure and initial values are defined by the core software, modules, and theme, but that you can edit. Examples include the labels for fields in your content types, header text in views, your site name, and the content of automatic email messages that your site sends out. After creating configuration text in the default language of your site, you can translate it into other languages. You need to have the core Configuration Translation module installed in order to translate this text.

Content text and files

If your site is multilingual, you can configure the content fields on your site to be translatable. After creating content in one language, you can translate it into other languages.

Fields can contain textual information or uploaded files, and for each field on each content type, you can configure it to be translatable or non-translatable. You need to have the core Content Translation module installed in order to translate this text.

Related topics

- [*Bagian 10.2, "Adding a Language"*](#)
- [*Bagian 10.3, "Configuring Content Translation"*](#)
- [*Bagian 10.4, "Translating Content"*](#)
- [*Bagian 10.5, "Translating Configuration"*](#)

Attributions

Written by [*Jennifer Hodgdon*](#).

10.2. Adding a Language

Goal

Add one or more languages to your site and define which one is used by default.

Prerequisite knowledge

[*Bagian 10.1, "Concept: User Interface, Configuration, and Content translation"*](#)

Steps

1. Install the four core multilingual modules (Language, Interface Translation, Content Translation, and Configuration Translation), by following the steps in [*Bagian 4.3, "Installing a Module"*](#).
2. In the *Manage* administrative menu, navigate to *Configuration > Regional and language > Languages* (`admin/config/regional/language`).
3. Click *Add language*.
4. Select *Spanish* (or your preferred language) from the *Language name* select list. Click *Add language*. After waiting for translations to finish downloading, you will be returned to the *Languages* page, with a confirmation message and the new language shown.

The screenshot shows the 'Languages' configuration page in Drupal. At the top, there are tabs for 'List' and 'Detection and selection'. Below the tabs, the breadcrumb navigation shows: Home > Administration > Configuration > Regional and language. A success message box contains the following text:
✓ The language *Spanish* has been created and can now be used.
Use one of the language switcher blocks to allow site visitors to switch between languages. You can enable these blocks on the [block administration page](#).
One translation file imported. 8057 translations were added, 0 translations were updated and 0 translations were removed.

Below the message box, there is a note: Reorder the configured languages to set their order in the language switcher block and, when editing content, in the list of selectable languages. This ordering does not impact [detection and selection](#).

The site default language can also be set. It is not recommended to change the default language on a working site. [Configure the Selected language setting](#) on the detection and selection page to change the fallback language for language selection.

Interface translations are automatically imported when a language is added, or when new modules or themes are enabled. The report [Available translation updates](#) shows the status. Interface text can be customized in the [user interface translation](#) page.

At the bottom left is a blue button labeled '+ Add language'. On the right, there is a link 'Show row weights'. The main table lists two languages:

NAME	DEFAULT	INTERFACE TRANSLATION	OPERATIONS
English	<input checked="" type="radio"/>	not applicable	<button>Edit</button>
Spanish	<input type="radio"/>	8057/8059 (99.98%)	<button>Edit</button>

At the bottom left of the table is a blue 'Save configuration' button.

5. Follow the steps in [Bagian 8.3, "Placing a Block in a Region"](#) to place the *Language switcher* block in the *Sidebar second* region. This will enable site visitors to switch between languages, once the site has been translated.

Expand your understanding

- [Bagian 10.3, "Configuring Content Translation"](#)
- [Bagian 10.4, "Translating Content"](#)

Additional resources

[Drupal.org page "Resource Guide: Configuring a Multilingual Site"](#)

Attributions

Written and edited by [Leila Tite](#), [Jennifer Hodgdon](#), and [Boris Doesborg](#).

10.3. Configuring Content Translation

Goal

Make *Custom block*, *Custom menu link*, and *Content* entity types translatable. Select specific subtypes and set which fields of these can be translated.

Prerequisite knowledge

- [Bagian 2.3, "Concept: Content Entities and Fields"](#)
- [Bagian 10.1, "Concept: User Interface, Configuration, and Content translation"](#)

Site prerequisites

The core Content Translation module must be installed, and your site must have at least two languages. See [Bagian 10.2, "Adding a Language"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Configuration > Regional and language > Content language and translation* ([admin/config/regional/content-language](#)).
2. Under *Custom language settings*, check *Content*, *Custom block* and *Custom menu link* to make these entity types translatable.

Custom language settings

- [Comment](#)
- [Contact message](#)
- [Content](#)
- [Custom block](#)
- [Custom menu link](#)
- [File](#)
- [Shortcut link](#)
- [Taxonomy term](#)
- [User](#)

Save configuration

3. Configuration options appear for *Content*, *Custom block* and *Custom menu link*. Choose the subtypes you want to translate for each entity type. Check *Basic page* for *Content*, *Basic block* for *Custom block* and *Custom menu link* for *Custom menu link*.
4. Verify the settings for the entity types as shown below:

Field name	Explanation	Example value
Default language	The default language for the entity subtype	Site's default language (English)
Show language selector on create and edit pages	Whether or not the language selector should be shown while editing and creating content	Checked

Default language

Basic block Explanation of the language options is found on the [languages list page](#).

Show language selector on create and edit pages

- Choose the fields that should be translatable for *Basic page* as shown in the table below. If a field is not translation-dependent, leave it unchecked. Enabling translation for fields that are numbers or dates has use for accountability or implementing workflows.

Field name	Explanation	Example value
Title	The title of the content	Checked
Authored by	The author	Unchecked
Publishing status	Whether the content has been published or not	Checked
Authored on	Date of publishing	Unchecked
Changed	Date of last update	Unchecked
Promoted to front page	Whether the content will be included in some content views	Unchecked
Sticky at top of lists	Whether the content will be displayed first in some content views	Unchecked
URL alias	Nicer URL for the content	Checked
Body	The main content of the page	Checked

- Title
- Authored by
- Publishing status
- Authored on
- Changed
- Promoted to front page
- Sticky at top of lists
- URL alias
- Body

6. Similarly, check the appropriate boxes for translatable fields belonging to *Basic block* and *Custom menu link*.
7. Click *Save configuration*.

Expand your understanding

- [Bagian 10.5, "Translating Configuration"](#)
- [Bagian 10.4, "Translating Content"](#)

Additional resources

- [Blog post "Multilingual Drupal 8 tidbits, part 5"](#)
- [Blog post "Multilingual Drupal 8 tidbits, part 17"](#)

Attributions

Written and edited by [Laura Vass](#) at [Pronovix](#), [Jojo Alphonso](#) at [Red Crackle](#), and [Jennifer Hoggdon](#).

10.4. Translating Content

Goal

Translate the home page to Spanish.

Prerequisite knowledge

[Bagian 10.1, "Concept: User Interface, Configuration, and Content translation"](#)

Site prerequisites

- The Home content item must exist. See [Bagian 5.2, "Creating a Content Item"](#).
- The core Content Translation module must be installed, and your site must have at least two languages. See [Bagian 10.2, "Adding a Language"](#).
- The *Basic page* content type must be configured to be translatable. See [Bagian 10.3, "Configuring Content Translation"](#).

Steps

1. In the *Manage* administrative menu, navigate to *Content* (`admin/content`).
2. Locate the home page. You can search for it by entering "Home" in the title field.
3. Select *Translate* from the dropdown button in the row of the Home content item. The page *Translations of Home* appears.
4. Click *Add* in the row *Spanish*.

The screenshot shows a table titled "Translations of Home" with a star icon. The table has columns: LANGUAGE, TRANSLATION, SOURCE LANGUAGE, STATUS, and OPERATIONS. There are two rows: one for "English (Original language)" with "Home" in the TRANSLATION column, and another for "Spanish" with "n/a" in the TRANSLATION column. The "Edit" button is visible for the English row, and the "Add" button is visible for the Spanish row. The URL in the browser is "Home > Node".

LANGUAGE	TRANSLATION	SOURCE LANGUAGE	STATUS	OPERATIONS
English (Original language)	Home	n/a	Published	Edit
Spanish	n/a	n/a	Not translated	Add

5. Note that the user interface has switched to Spanish. To switch it back to English, remove the first instance of `es` in the browser's URL. For example, if your URL looks like `example.com/es/node/5/translations/add/en/es`, remove the `es` that comes immediately after `example.com`.
6. Fill in the fields as shown below. You will have to click and expand *URL path settings* on the right to provide the value for *URL alias*.

Field name	Explanation	Value
Title	Translated title of the page	Página principal

Field name	Explanation	Value
Body	Translated body of the page	Bienvenido al mercado de la ciudad - ¡el mercado de agricultores de tu barrio! Horario: Domingos de 9:00 a 14:00. Desde Abril a Septiembre Lugar: parking del Banco Trust número 1. En el centro de la ciudad
URL path settings > URL alias	Translated address of the webpage	pagina-principal

7. Click *Save and keep published (this translation)*.
8. Go to your site's home page to view the newly translated page.

Expand your understanding

- Follow the steps above to translate more content on your site.
- [*Bagian 10.5, "Translating Configuration"*](#)

Attributions

Written by [*Boris Doesborg*](#).

10.5. Translating Configuration

Goal

Translate the labels on the Recipes view page into Spanish.

Prerequisite knowledge

- [*Bagian 10.1, "Concept: User Interface, Configuration, and Content translation"*](#)
- [*Bagian 9.3, "Creating a Content List View"*](#)

Site prerequisites

- The core Configuration Translation module must be installed, and your site must have at least two languages. See [*Bagian 10.2, "Adding a Language"*](#).
- The Recipes view must exist. See [*Bagian 9.3, "Creating a Content List View"*](#) and [*Bagian 9.4, "Duplicating a View"*](#).

Steps

The basic steps for translating any configuration on your site are:

1. Locate the page where the configuration you want to translate is created or edited in the site's primary language. For example, to translate the site name, you need to find the page where you would edit the site name (see [Bagian 4.2, "Editing Basic Site Information"](#)).
2. Find a link, tab, or button on the page that says "Translate" or something similar. Click this link, tab, or button.
3. Find a button that will let you add a translation in the desired language, and click this button.
4. Enter the translation in the form, and save.

Most configuration is fairly straightforward and intuitive to edit in this manner. Views configuration is an exception, because the translation editing form is nothing like the view editing form, and it is complex and hierarchical rather than being a simple form with just a few fields. As an example of how to translate a view, here are the steps to translate the labels in the Recipes view to Spanish:

1. In the *Manage* administrative menu, navigate to *Structure > Views* (`admin/structure/views`).
2. Locate the Recipes view and click *Translate* from the dropdown button.
3. Click *Add* in the row *Spanish*. The page *Add Spanish translation for Recipes view* appears.
4. Find *Displays > Master Display settings > Recipes default display Options*. Translate the *Display title* from "Recipes" to "Recetas".
5. Find *Displays > Master display settings > Recipes default display options > Exposed form > Reset options*. Translate *Submit button text (Apply)* to "Aplicar". The other buttons and labels in this section do not appear on the Recipes page or block, and do not need to be translated.

Add Spanish translation for Recipes view ☆

Home > Administration > Structure > Views > Recipes > Translate

Label Recipes	Label Recipes
Administrative description (Empty)	Administrative description

DISPLAYS

MASTER DISPLAY SETTINGS

Title Master	Title Master
------------------------	------------------------

RECIPES DEFAULT DISPLAY OPTIONS

EXPOSED FORM

RESET OPTIONS	
Submit button text Apply	Submit button text Apply
Reset button label Reset	Reset button label Reset
Exposed sorts label Sort by	Exposed sorts label Sort by

6. Find *Displays > Master display settings > Recipes default display options > Filters > (Empty) taxonomy term ID > Find recipes using... Expose*. In the *Label* field, enter "Encontrar recetas usando...".
7. Click *Save translation*.
8. Navigate to the Recipes page and switch to Spanish using the Language switcher block. Verify that the labels have been translated.

Expand your understanding

- Translate the block display title in the Recent recipes display settings section of the Recipes view.
- Translate the page title in the Vendors view.
- Translate other configuration. Some examples of where to find the translation pages:
 - To translate the site name, navigate in the *Manage* administrative menu to *Configuration > System > Basic site settings > Translate system information* (`admin/config/system/site-information/translate`).
 - To translate the contact form, navigate in the *Manage* administrative menu to *Structure > Contact forms* (`admin/structure/contact`). Click *Translate* in the dropdown button in the *Website feedback* row.
 - To translate the name of a menu, navigate in the *Manage* administrative menu to *Structure > Menus* (`admin/structure/menu`). Click *Translate* in the dropdown button for the menu whose name you want to translate.

- Menu items within a menu are considered to be content (not configuration); see [Bagian 10.3, "Configuring Content Translation"](#) to enable translation. Once translation is enabled, navigate in the *Manage* administrative menu to *Structure > Menus* (*admin/structure/menu*). Click *Edit menu* in the dropdown button for the menu whose links you want to translate. Click *Translate* in the dropdown button for the link you want to translate.
- To translate field labels on a content type, navigate in the *Manage* administrative menu to *Structure > Content types* (*admin/structure/types*). Click *Manage fields* in the dropdown button for the content type whose field labels you want to edit. Click *Translate* in the dropdown button for the field whose label you want to edit.
- Translate content. See [Bagian 10.4, "Translating Content"](#).

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

Bab 11. Mengembangkan dan Menyesuaikan Situs Anda

11.1. Finding Modules

Goal

Find and evaluate modules on *Drupal.org*.

Prerequisite knowledge

- [Bagian 1.1, "Concept: Drupal as a Content Management System"](#)
- [Bagian 1.2, "Concept: Modules"](#)

Steps

1. Go to [Drupal.org](#), and navigate to *Download & Extend > Modules* (https://www.drupal.org/project/project_module).
2. Filter your search using the categories on the module search page. Fill in the fields as shown below.

Field name	Explanation	Example value
Maintenance status	How actively should the module be maintained?	Actively maintained
Development status	What kind of development should the module undergo?	Any
Module categories	The module's topic area.	Administration
Core compatibility	The version of the core software the module is compatible with.	8.x
Status	Project status: <i>Sandbox projects</i> are experimental projects. <i>Full projects</i> have already gone through an approval process, but they can still be in development.	Full projects
Search Modules	Search for <i>Admin Toolbar</i> , a module that will be covered	Admin Toolbar

Field name	Explanation	Example value
	in detail later. Alternatively, you can also leave the field blank if you are not sure which module to search for.	
Sort by	Order your search results by criteria like <i>Most installed</i> (popular modules that many sites use) or <i>Last release</i> (date of latest version released).	Most installed

46 Modules match your search

Maintenance status - Any ▾

Development status - Any ▾

Module categories - Any ▾

Core compatibility 8.x ▾

Status Full projects ▾

Stability All projects ▾

Search Modules Admin Toolbar

Sort by Most installed ▾

Search

- Click *Search*. Search results will appear.

Admin Toolbar

Posted by eme on 20 April 2015

What is Admin Toolbar module ?

Admin Toolbar intends to improve the default Drupal Toolbar (the administration menu at the top of your site) to transform it into a drop-down menu, providing a fast access to all administration pages.

The module works on the top of the default toolbar core module and is therefore a very light module and keeps all the toolbar functionalities (shortcut / media responsive).

How to use Admin Toolbar module ?

Just install it like any other module.

[Read more](#) · Categories: Actively maintained , Under active development , Administration

Adminimal Admin Toolbar (Drupal 8)

Posted by energie on 19 March 2016

This module is inspired by [Adminimal Administration Menu](#). It provides a minimalist style to [Admin Toolbar](#) for Drupal 8.

Although the "Adminimal" theme provides menu styling, there are some conflicts with Admin toolbar and styling will only be present when the admin theme is set like when editing content.

[Read more](#) · Categories: Actively maintained , Under active development , Administration ,

Theme Enhancements

Toolbar Themes

Posted by Jeff Burnz on 13 September 2016

Provides themes for the Drupal 8 Toolbar module. The main purpose of this module is to provide compact, minimal themes for the Toolbar and allow the site admin to show or hide tabs and icons.

Installation

Install like all other modules, then go to: [~/admin/config/toolbar-themes/settings](#)

[Read more](#) · Categories: Actively maintained , Under active development , Administration

4. To further evaluate a module, click its title in the list of search results to visit its project page.

Some aspects to pay attention to when evaluating modules:

- Project description: The description of the module on its project page should be clear and useful. You should get an idea of its features and requirements.
- Maintenance status: If a module is *Actively maintained*, you can be sure that there will be security updates, bug fixes and feature improvements provided on a regular basis. However, if the module is unmaintained or abandoned, you shouldn't use it.

- Development status: *Under active development* means you can expect new features to be added to the project, but some aspects (for example, API) may still change. *Maintenance fixes only* means that the project is considered feature complete.
- Reported installs, downloads: You can see how many people have downloaded and how many sites use the module. If it's only used by a few sites, it might be a unique solution that not many people need, or it might be a warning sign that you shouldn't use it either.
- Maintainers: When was the last commit (the last time someone updated something on the module) or last release (new version)? If the project has few open issues, a long time since commits/releases might be appropriate, but if it has a lot of open issues and there are no commits/releases, that would be a clue that it might be abandoned.
- Issues: See if there are any open issues, potential problems with the module. Check the *Statistics* to see how regularly issues are responded to.
- Resources: Check if the module has documentation or a README file, that can help you install, configure, explore, and test it.

Admin Toolbar

[View](#) [Version control](#) [Automated Testing](#)

Posted by [eme](#) on April 20, 2015 at 2:28pm

Admin Toolbar

What is Admin Toolbar module ?

Admin Toolbar intends to improve the default Drupal Toolbar (the administration menu at the top of your site) to transform it into a drop-down menu, providing a fast access to all administration pages.

The module works on the top of the default toolbar core module and is therefore a very light module and keeps all the toolbar functionalities (shortcut / media responsive).

How to use Admin Toolbar module ?

Just install it like any other module.

What more?

The module provides also a submodule called "Admin Toolbar Extra Tools" adding extra links like what does Admin Menu module for Drupal 7 (flush caches, run CRON, etc...).

Please provide feedback and ideas!

Complementary modules :

[Toolbar Menu](#) allows you to add easily any menu to your toolbar and plays nicely with Admin toolbar making them dropdown menus.

[Toolbar Anti-flicker](#) removes the impact of the admin toolbar sub-menu on the rest of the page.

[Coffee](#) is a good module to look for admin paths.

[Adminimal Admin toolbar](#) provides a black theming, recommended for the [Adminimal theme](#).

Supporting organizations:

[emerya](#)

Project Information

Maintenance status: [Actively maintained](#)

Development status: [Under active development](#)

Module categories: [Administration](#)

Reported installs: **14,813** sites currently report using this module. [View usage statistics](#).

Downloads: 81,993

Automated tests: Enabled

Last modified: August 9, 2016

shield [Stable releases receive coverage from the Drupal Security Team](#).

Look for the shield icon below.

Maintainers for Admin Toolbar

[romainj](#) - 29 commits

last: 5 days ago, first: 4 months ago

[eme](#) - 11 commits

last: 3 weeks ago, first: 8 months ago

[matio89](#) - 23 commits

last: 2 months ago, first: 1 year ago

[View all committers](#)

[View commits](#)

Issues for Admin Toolbar

To avoid duplicates, please search before submitting a new issue.

[Search](#)

[Advanced search](#)

[All issues](#)

[15 open, 138 total](#)

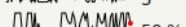
[Bug report](#)

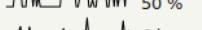
[6 open, 78 total](#)

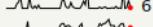
[Subscribe via e-mail](#)

[Statistics](#)

New issues  3

Response rate  50 %

1st response  6 hours

Open bugs  6

Participants  7

[2 year graph, updates weekly](#)

Resources

[Read license](#)

Expand your understanding

[Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#)

Attributions

Written by [Diána Lakatos](#) at [Pronovix](#).

11.2. Enabling and Disabling Maintenance Mode

Goal

Put your site in maintenance mode to allow users with the right permissions to use the site while users without this permission are presented with a message that the site is under maintenance.

Prerequisite knowledge

[Bagian 13.3, "Concept: Security and Regular Updates"](#)

Steps

Enabling maintenance mode

1. In the *Manage* administrative menu, navigate to *Configuration > Development > Maintenance mode* (`admin/config/development/maintenance`). The *Maintenance mode* page appears.
2. Fill in the fields as shown below.

Field name	Explanation	Value
Put site into maintenance mode	Enable the maintenance mode	Checked
Message to display when in maintenance mode	The information that is shown to website visitors when the mode is enabled. Variables such as @site can be used in the message	@site is currently under maintenance but should be back shortly. Thank you for your patience.

3. Click *Save configuration*.
4. Verify that the site is in maintenance mode by accessing it from another browser where you aren't logged in. If you are not able to verify, try clearing the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Anytown Farmers Market
Farm Fresh Food

Site under maintenance

Anytown Farmers Market is currently under maintenance. We should be back shortly. Thank you for your patience.

Disabling maintenance mode

1. In the *Manage* administrative menu, navigate to *Configuration > Development > Maintenance mode* (`admin/config/development/maintenance`). The *Maintenance mode* page appears.
2. Fill in the fields as shown below.

Field name	Explanation	Value
Put site into maintenance mode	Disable the maintenance mode	Unchecked
Message to display when in maintenance mode	No message required while disabling. You can leave the field blank.	

3. Click *Save configuration*.
4. Verify that the site is no longer in maintenance mode by accessing it from another browser where you aren't logged in. If you are not able to verify, try clearing the cache. See [Bagian 12.2, "Clearing the Cache"](#).

The screenshot shows the homepage of the Anytown Farmers Market website. At the top left is a circular logo with "ANYTOWN FARMERS MARKET" around the top and "FARM FRESH FOOD" around the bottom. Inside the circle is a stylized illustration of a farm with a house, barn, and fields. To the right of the logo is the text "Anytown Farmers Market" and "Farm Fresh Food". Below the logo is a navigation bar with four buttons: "Home", "About", "Recipes", and "Vendors". The main content area has a green gradient background. The word "Home" is bolded. Below it is the text "Welcome to City Market - your neighborhood farmers market!". Underneath that are three lines of information: "Open: Sundays, 9 AM to 2 PM, April to September", "Location: Parking lot of Trust Bank, 1st & Union, downtown", and "Hours and location". To the right of this is a box titled "Language" with "English" and "Español" options. Another box titled "New recipes" shows two images: one of carrots and another of a salad.

Expand your understanding

- [Bagian 13.5, "Updating the Core Software"](#)
- [Bagian 13.7, "Updating a Theme"](#)
- [Bagian 13.6, "Updating a Module"](#)

Attributions

Written and edited by [Boris Doesborg](#), and [Jojo Alphonso](#) at [Red Crackle](#).

11.3. Downloading and Installing a Module from *Drupal.org*

Goal

Download and install the [contributed Admin Toolbar module](#), which allows you to easily browse through the administration section of the website.

Prerequisite knowledge

- [Bagian 1.2, "Concept: Modules"](#)
- [Bagian 11.1, "Finding Modules"](#)
- [Bagian 3.2, "Concept: Additional Tools"](#)

Site prerequisites

- If you want to install modules via the website, the core Update Manager module must be installed. See [Bagian 4.3, "Installing a Module"](#) for instructions on installing modules.
- If you want to use Drush, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

You can use the administrative interface or Drush to install a contributed module. If you are installing a custom module rather than a contributed module, if you see a message saying *Installing modules and themes requires FTP access to your server*, or if the steps below do not work to download and unpack the module files, follow the steps in [Bagian 11.6, "Manually Downloading Module or Theme Files"](#). If you are installing a contributed module with external dependencies that are managed by Composer, follow the steps for downloading in [Bagian 11.9, "Using Composer and Git to Download Files"](#). In either of these cases, then continue here with step 7 in the instructions for the administrative interface below.

Using the administrative interface

1. On the *Admin toolbar* project page on drupal.org (https://www.drupal.org/project/admin_toolbar), scroll to the *Downloads* section at the bottom of the page.
2. Copy the address of the *tar.gz* link. Depending on your device and browser, you might do this by right clicking and selecting *Copy link address*.

Downloads

Recommended releases

Version	Download	Date
8.x-1.16	tar.gz (55.55 KB) zip (62.65 KB)	2016-Jul-28

Development releases

Version	Download	Date
8.x-1.x-dev	tar.gz (16.89 KB) zip (26.26 KB)	2016-Aug-19

[View all releases](#)

3. In the *Manage* administrative menu, navigate to *Extend (admin/modules)*. The *Extend* page appears.
4. Click *Install new module*. The *Install new module* page appears.

Install new module ★

Home » Administration » Extend

You can find [modules](#) and [themes](#) on [drupal.org](#). The following file extensions are supported: zip tar tgz gz bz2.

Install from a URL

For example: <http://ftp.drupal.org/files/projects/name.tar.gz>

Or

Upload a module or theme archive to install

[Browse...](#) No file selected.

For example: name.tar.gz from your local computer

[Install](#)

5. In the field *Install from a URL*, paste the copied download link. This value could look like this: https://ftp.drupal.org/files/projects/admin_toolbar-8.x-1.14.tar.gz
6. Click *Install* to upload and unpack the new module on the server. The files are being downloaded to the *modules* directory.
7. Click *Enable newly added modules* to return to the *Extend* page. If you used the manual uploading procedure, start with this step, and reach the *Extend* page by using the *Manage* administrative menu and navigating to *Extend (admin/modules)*.
8. Locate and check *Admin toolbar*.
9. Click *Install* to turn on the new module.

Using Drush

1. Find the project name for the module you want to install, which is the last segment of the module's project page URL. For example, if the project URL is https://www.drupal.org/project/admin_toolbar, the project name is "admin_toolbar".
2. Run the following Drush commands, giving the project name (for example, admin_toolbar) as a parameter:

```
drush dl admin_toolbar  
drush en admin_toolbar
```

3. Follow the instructions on the screen.

Expand your understanding

- Verify that the [*contributed Admin Toolbar module*](#) is working by browsing through the menu in the administration section.
- Install and configure the [*contributed Pathauto module*](#) so that content pages in your site get nice URLs by default. See [*Bagian 5.1, "Concept: Paths, Aliases, and URLs*](#) for more on URLs.

- If you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Additional resources

- [Drupal.org community documentation page "Contributed modules"](#)
- ["Download and Extend" page on Drupal.org](#)
- [Admin Toolbar module on Drupal.org](#)

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

11.4. Finding Themes

Goal

Find and evaluate themes on *Drupal.org*.

Prerequisite knowledge

- [Bagian 1.1, "Concept: Drupal as a Content Management System"](#)
- [Bagian 1.3, "Concept: Themes"](#)

Steps

1. Go to <https://www.drupal.org>, and navigate to *Download & Extend > Themes* (https://www.drupal.org/project/project_theme).
2. Filter your search using the categories on the theme search page. For example, you might use these filters:

Field name	Explanation	Example value
Maintenance status	How actively is the theme maintained? If a theme is actively maintained, you can expect bug fixes and improvements on a regular basis.	Actively maintained
Development status	What kind of development is the theme undergoing? If you select <i>Under active de-</i>	Any

Field name	Explanation	Example value
	<i>velopment</i> , you can expect new features to be added, and some aspects may still change. If you select <i>Maintenance fixes only</i> , it means that the theme is considered complete.	
Core compatibility	The Drupal version the theme is compatible with.	8.x
Status	<i>Sandbox projects</i> are experimental projects. <i>Full projects</i> have already gone through an approval process, but they can still be in development.	Full projects
Search Themes	Search by search term in the theme's description.	-
Sort by	Order your search results by criteria like <i>Most installed</i> (popular themes that many sites use) or <i>Last release</i> (date of latest version released).	Most installed

179 Themes match your search

Maintenance status: Actively maintained | ▾

Development status: - Any - | ▾

Core compatibility: 8.x | ▾

Status: Full projects | ▾

Stability: All projects | ▾

Search Themes:

Sort by: Most installed | ▾

Search

- Click **Search**. Search results will appear.

Bootstrap

Posted by [wundo](#) on 18 May 2008

“ Sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. Bootstrap has become one of the most popular front-end frameworks and open source projects in the world.

This base theme bridges the gap between Drupal and the [Bootstrap Framework](#).

Features

- [jsDelivr CDN](#) for "out-of-the-box" styling and faster page load times.
- [Bootswatch](#) theme support, if using the CDN.
- Glyphicons support via [Icon API](#).
- Extensive integration and template/preprocessor overrides for most of the [Bootstrap Framework](#) CSS, Components and JavaScript
- Theme settings to further enhance the Drupal Bootstrap integration:
 - [Breadcrumbs](#)
 - [Navbar](#)
 - [Popovers](#)
 - [Tooltips](#)
 - [Wells](#) (per region)

Documentation

Visit the project's [official documentation site](#) or the markdown files inside the [./docs](#) directory.

Supported modules

Drupal 8

- [Bootstrap Layouts](#)

Drupal 7

- [Bootstrap Core](#)
- [jQuery Update](#)
- [Icon API](#)
- [Picture](#)
- [Views \(partial support\)](#)
- [Webform \(partial support\)](#)

5 Year Evolution ([source](#))

<https://youtu.be/Cvq6MPJp2dI>

4. To further evaluate a theme, click its title in the list of search results to visit its project page.

Some aspects to pay attention to while evaluating themes:

- Introduction: The description of the theme on its project page should be clear and useful. A screenshot of the theme helps your evaluation as well.
- Downloads: The theme should be available for your version of the core software. Pay attention to the color-coding: Themes marked green are recommended to be used on live

sites, the ones marked yellow should be carefully evaluated before use, and the ones marked red are only recommended for testing.

- Project Information: Check the theme's maintenance and development status, and how frequently it's downloaded and installed.
- Issues: See if there are any open issues or potential problems with the theme.
- Resources: Check if the theme has documentation.

Expand your understanding

- [*Bagian 11.5, "Downloading and Installing a Theme from Drupal.org"*](#)

Attributions

Written by [*Diána Lakatos*](#).

11.5. Downloading and Installing a Theme from *Drupal.org*

Goal

Download and install a theme from *Drupal.org*.

Prerequisite knowledge

- [*Bagian 11.4, "Finding Themes"*](#)
- [*Bagian 3.2, "Concept: Additional Tools"*](#)

Site prerequisites

If you want to install via the website, the core Update Manager module must be installed. See [*Bagian 4.3, "Installing a Module"*](#) for instructions on installing modules.

If you want to use Drush, Drush must be installed. See [*Bagian 3.2, "Concept: Additional Tools"*](#).

Steps

You can use the administrative interface or Drush to install a contributed theme. If you are installing a custom theme rather than a contributed theme, if you see a message saying *Installing modules and themes requires FTP access to your server*, or if the steps below do not work to download and unpack the theme files, follow the steps in [*Bagian 11.6, "Manually Downloading Module or Theme Files"*](#). If you are installing a contributed theme with external dependencies that are managed by Composer, follow the steps for downloading in [*Bagian 11.9*](#).

["Using Composer and Git to Download Files"](#). In either of these cases, then continue here with step 7 in the instructions for the administrative interface below.

Using the administrative interface

1. On the theme's project page on drupal.org (for example, <https://www.drupal.org/project/mayo>), scroll to the *Downloads* section at the bottom of the page.
2. Right-click *tar.gz* to copy the address.

Downloads

Version	Download	Date
8.x-1.3	tar.gz (437.31 KB) zip (486.26 KB)	2016-May-09
7.x-2.6	tar.gz (411.91 KB) zip (441.6 KB)	2015-Nov-11
7.x-1.4	tar.gz (386.35 KB) zip (403.15 KB)	2015-Nov-11

Development releases

8.x-1.x-dev	tar.gz (437.31 KB) zip (486.27 KB)	2016-Apr-27
7.x-2.x-dev	tar.gz (411.92 KB) zip (441.62 KB)	2016-Feb-11

[View all releases](#)

3. In the *Manage* administrative menu, navigate to *Appearance* ([admin/appearance](#)). The *Appearance* page appears.
4. Click *Install new theme*. The *Install new theme* page appears.

Install new theme ☆

Home » Administration

You can find [modules](#) and [themes](#) on [drupal.org](#). The following file extensions are supported: `zip tar tgz gz bz2`.

Install from a URL

For example: <http://ftp.drupal.org/files/projects/name.tar.gz>

Or

Upload a module or theme archive to install

No file selected.

For example: `name.tar.gz` from your local computer

5. In the field *Install from a URL*, paste the copied download link. This value could look like `https://ftp.drupal.org/files/projects/mayo-8.x-1.3.tar.gz`.
6. Click *Install* to upload and unpack the new theme on the server. The files are being downloaded to the *themes* directory.
7. Click *Install newly added themes* to return to the *Appearance* page. If you used the manual uploading procedure, start with this step, and reach the *Appearance* page by using the *Manage* administrative menu and navigating to *Appearance* ([admin/appearance](#)).

8. Locate the new theme under *Uninstalled themes* and click *Install and set as default* to use it. All non-administrative pages on the site will now use this new theme.



Mayo 8.x-1.3

Simple but fully customizable and colorable responsive layout theme.

[Install](#) | [Install and set as default](#)

Using Drush

1. Find the project name for the theme you want to install, which is the last segment of the theme's project page URL. For example, if the project URL is <https://www.drupal.org/project/mayo>, the project name is `mayo`.
2. Run the following Drush commands, giving the project name (for example, `mayo`) as a parameter:

```
drush dl mayo  
drush en mayo  
drush config-set system.theme default mayo
```

3. Follow the instructions on the screen.

Expand your understanding

- In the *Manage* administrative menu, navigate to *Appearance (admin/appearance)* and uninstall any themes that you are not using.
- [Bagian 11.1, "Finding Modules"](#)
- [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#)
- If you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Attributions

Written and edited by [Joe Shindelar](#) at [Drupalize.Me](#), and [Boris Doesborgh](#).

11.6. Manually Downloading Module or Theme Files

Goal

Manually download module or theme files and upload them to your site, if the website or Drush methods for installing or updating a module or theme do not work, or if you are placing a custom-written module or theme.

Prerequisite knowledge

- [Bagian 1.2, "Concept: Modules"](#)
- [Bagian 11.1, "Finding Modules"](#)
- [Bagian 1.3, "Concept: Themes"](#)
- [Bagian 11.4, "Finding Themes"](#)

Site prerequisites

You need to be facing any of the following to perform the manual download described in this topic:

- File permission issues
- FTP permission issues
- You created a custom module/theme or received its files from someone
- You could not successfully complete the instructions in [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#), [Bagian 11.5, "Downloading and Installing a Theme from Drupal.org"](#), [Bagian 13.6, "Updating a Module"](#), or [Bagian 13.7, "Updating a Theme"](#)

Skip this topic if none of the above applies to you.

Steps

If you are installing a module or theme from *Drupal.org*, follow the downloading instructions, and then the uploading/unpacking instructions. If you created the module or theme, skip the downloading step. Instead, create an archive file (that you know how to extract on the server) and proceed with the steps in uploading/unpacking, using whatever method is appropriate for the way you initially created the archive file.

Downloading the files

1. Open the module or theme project page on drupal.org; for example, the *Admin toolbar* page (https://www.drupal.org/project/admin_toolbar).
2. Scroll to the *Downloads* section near the bottom of the page.

Downloads

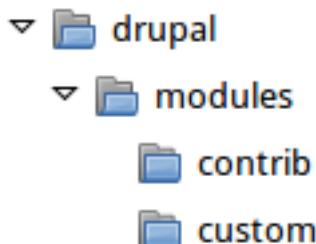
Version	Download	Date
8.x-1.19	tar.gz (17.07 KB) zip (27.02 KB)	2017-Apr-06
Development releases		
8.x-1.x-dev	tar.gz (19.49 KB) zip (29.81 KB)	2017-Jun-11

[View all releases](#)

- Click *tar.gz* to download the file to your computer.

Uploading the files to your site and unpacking them

- If you are adding a new module or theme, create subdirectories in your top-level *modules* and *themes* directories (if they don't already exist). Typically, people make a *contrib* subdirectory for contributed modules and themes that are downloaded from *Drupal.org*, and a *custom* subdirectory for modules and themes that they created. Your *modules* directory might look like this:



- If you are replacing an existing module or theme with an updated version, put the site into maintenance mode. See [Bagian 11.2, "Enabling and Disabling Maintenance Mode"](#).
- If you are replacing an existing module or theme with an updated version, find and delete all the existing files and directories for the existing module or theme. Modules are normally located in directories under the top-level *modules* directory, and themes are normally located in directories under the top-level *themes* directory.
- Upload the *.tar.gz* file (or whatever archive you created) to your site. Place it in either the same location from which you deleted the directory (if replacing an existing module or theme) or the appropriate subdirectory of *modules* or *themes* (if adding a new module or theme).
- Extract the files from the *.tar.gz* archive (or whatever archive you created), making a subdirectory in the same location as the archive file. If you do not have terminal access, or your hosting server is not running Linux, your hosting control panel's file manager should provide a way to extract the files. If you have terminal access to your hosting server (running Linux) and you are using a tar.gz file, you can use a command like:

```
tar -xzf admin_toolbar-8.x-1.17.tar.gz
```

- Delete the compressed file from the server, unless your unpacking method already deleted it.
- Refer to [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#), [Bagian 11.5, "Downloading and Installing a Theme from Drupal.org"](#), [Bagian 13.6, "Updating a Module"](#),

or [Bagian 13.7, "Updating a Theme"](#) to complete the installation or update of the module or theme. Start at the step after the automatic download has been completed.

Expand your understanding

- If you work with multiple environments (for example, a local development site and a production site) you will have to repeat the steps on each environment, or re-clone the environment. See [Bagian 11.8, "Making a Development Site"](#).
- If you added a new theme, navigate in the *Manage* administrative menu to *Appearance* (*admin/appearance*) and uninstall the old theme.

Additional resources

- [Drupal.org community documentation page "Updating modules"](#)
- [Drupal.org community documentation page "Installing contributed modules \(Drupal 8\)"](#)
- [Drupal.org community documentation page "Installing themes"](#)

Attributions

Written and edited by [Boris Doesborg](#), [Jennifer Hodgdon](#), and [Marc Isaacson](#).

11.7. Concept: Development Sites

What are Development Sites?

Development Sites are different copies of the same site used for developing, updating, and testing a site without risking the integrity of the live site.

An example deployment workflow for site building will usually include the sites mentioned below:

Local environment

The development process starts with developers working on new features, bug fixes, theming, and configuration in their local environment.

Development site

Developers push the changes they've been working on to the development site. For a team of more than one developer, version control is usually used. Git is a version control system that tracks your files for any changes. You can then commit those changes to a repository. Using Git allows team members to work on the same site without overriding each other's work. It also makes it possible to easily roll back to previous stages of the development.

Staging site

The staging site can be used for testing, or presenting the changes to the client for approval. QA (Quality Assurance) and UAT (User Acceptance Testing) are most often carried out on the staging site. It is recommended to have live content on both the development and staging sites, so that you can test how the new features will work with the existing content.

Production site

The live site on the web available to visitors. It contains new features that have been proven safe to go live.

Based on the project's size, scope, requirements, or stakeholders, stages from the above workflow can be removed, or additional stages can be added. For example, a testing site before staging can be added to separate testing and user acceptance processes.

Related topics

- [*Bagian 11.8, "Making a Development Site"*](#)
- [*Bagian 2.6, "Concept: Editorial Workflow"*](#)

Additional resources

[*Drupal.org community documentation page "Introduction to Git"*](#)

Attributions

Written and edited by [*Diána Lakatos*](#), and [*Joji Alphonso*](#) at [*Red Crackle*](#).

11.8. Making a Development Site

Goal

Make a copy of a site that you can use to develop new features and test updates on.

Prerequisite knowledge

[*Bagian 11.7, "Concept: Development Sites"*](#)

Site prerequisites

- You have a live, developed site that you would like to make a copy of for development purposes.

- If you want to use Drush for some of the steps in this task, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

1. Follow the steps in [Bagian 3.3, "Preparing to Install"](#), so that you have hosting set up for your development site, you know where the web root is for your development site, and you have an empty database and database user for your development site to use.
2. Make a database dump file from your live site's database. Try one of the following methods:
 - If you are using MySQL as your database, and your live site's server has phpMyAdmin installed (it is available from many hosting control panels), use the *Export* tab in phpMyAdmin to export in SQL format. Using *gzip* compression on the output file is suggested, to reduce the file size.
 - If you are using MySQL and have access to the command line, use this command (substituting in your site's database name, user name, and password):

```
mysqldump -u USERNAME -p'PASSWORD' DATABASENAME > BACKUPFILE.sql
```

- If you prefer to use Drush, use this command:

```
drush sql-dump > BACKUPFILE.sql
```

- Use the [contributed Backup and Migrate module](#) from within your live site. See [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#) for instructions on installing contributed modules.

You now have a database dump stored in the file *BACKUPFILE.sql*. For security reasons, avoid storing this file on your hosting server anywhere under the Drupal site root. This will prevent others from getting a copy of your database.

3. Copy all of the files from the web root of your live site to the web root of your development site.
4. Edit the *sites/default/settings.php* file under your development site's top-level directory in a plain-text editor. Find the lines near the end that contain the database name, database username, and database password, and update them to the information about the development site database you set up. The lines look something like this (before editing):

```
$databases['default']['default'] = array (
  'database' => 'live_site_database_name',
  'username' => 'live_site_database_username',
  'password' => 'live_site_database_password',
```

5. Check whether your *settings.php* file has the following setting; if yes, then you will need to edit this to point to your development site URL instead of your production site URL:

```
$settings['trusted_host_patterns']
```

6. Import the database dump file you created, into the development site's database. Try one of the following methods:

- If you are using MySQL as your database, and your live site's server has PHPMyAdmin installed (it is available from many hosting control panels), use the *Import* tab in PHPMyAdmin. You may find that you have to restart the import a few times, if your database was large.
- If you are using MySQL and have access to the command line, use this command (substituting in your site's database name, user name, and password; if you made a gzip-compressed backup file, you will also need to uncompress it first):

```
mysql -u USERNAME -pPASSWORD DATABASE_NAME < BACKUPFILE.sql
```

- If you prefer to use Drush, use this command:

```
drush sql-query --file=BACKUPFILE.sql
```

7. If your development and live sites need to have different configuration, then you have to use configuration overrides in the *settings.php* file. The *\$config* variable will help you maintain override values separately from the standard configuration data. For instance, you might want the site name to be "Anytown Farmers Market" on the production site, but "Development Site for Anytown Farmers Market" on the development site. To do that, you could have the production value in the site configuration (in the database), and on the development site, in the *settings.php* file, you would need to have:

```
$config['system.site'][name] = "Development Site for Anytown Farmers Market";
```

Expand your understanding

- Verify that the development site is working correctly.
- Log in to the development site as an administrator, and clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).
- [Bagian 11.10, "Deploying New Site Features"](#)

Additional resources

- [Drupal.org community documentation page "Backing up a site"](#)
- [Drupal.org community documentation page "Create a Test Site"](#)

Attributions

Written and edited by [Jennifer Hodgdon](#), [Joe Shindelar](#) at [Drupalize.Me](#), and [Jojo Alphonso](#) at [Red Crackle](#).

11.9. Using Composer and Git to Download Files

Goal

Use Composer and/or Git to download files and dependencies in the core software, or in add-on modules and themes.

Prerequisite knowledge

- [Bagian 3.2, "Concept: Additional Tools"](#)
- [Bagian 11.7, "Concept: Development Sites"](#)

Site prerequisites

If you want to use Git or Composer, they must be installed either on a local development server or your live site. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

If you are unable to install the Git or Composer tools on your live server, after following the steps in any of the sections below on your local server, you will need to transfer any updated or added files to your live server. The recommended procedure is to make an archive or zip file of the new and changed files, transfer the archive to your live server, and extract it there. If you are using Composer, make sure to check for updates and additions to the following files, in the root of your installation:

- *vendor* directory
- *autoload.php*
- *composer.json*
- *composer.lock*

Using Git to download a project

Follow these steps if you want to download the development version of the core software, or of a contributed module or theme, from the project's Git version control repository:

1. In a command-line window, change to one level above the directory where you want the files to reside. For the core software, this would be your web root. For an add-on module, this would be the *modules* directory, or a sub-directory like *contrib* under the *modules* directory.
2. Locate the page for the project that you want to download on [Drupal.org](https://www.drupal.org); for example, <https://www.drupal.org/project/drupal> for the core software, or https://www.drupal.org/project/admin_toolbar for the contributed Admin Toolbar module.
3. Click the *Version control* link near the top of the page, which opens up a page giving Git commands for this project.
4. Copy the `git clone` command under *Setting up repository for the first time*, and paste it into your command line. For example, it might be:

```
git clone --branch 8.2.x https://git.drupal.org/project/drupal.git
```

5. This will create a directory named with the short name of the project (*drupal* in this example), and download the required files into that directory. If necessary, rename the directory.

6. If you downloaded the *drupal* project (the core software), follow the instructions below to download its external dependencies.
7. Later on, you can change to the created directory and run the command `git pull` to update the files to the latest version from the Git repository for the project.

Using Composer to download external core dependencies

Follow these steps if you downloaded the core software files using Git, which does not include the external dependencies:

1. Change to the core software's root directory.
2. Run the command `composer install` at the command line. The external dependencies will be downloaded.

Using Composer to download the core software

Follow these steps if you have not yet downloaded or installed the core software, and you want to use Composer to download both the core software and its external dependencies:

1. At the command line, change to one level above the directory where you want the software to reside.
2. Enter this command, where `my_site_name` is the directory you want to create:

```
composer create-project drupal/drupal my_site_name
```

Using Composer to download a module or theme

Follow these steps if you have already downloaded the core software, and you want to use Composer to add a contributed module or theme (usually because it has external dependencies that need to be downloaded with Composer):

1. If you have not already done so, tell Composer about the download location for contributed modules and themes, by entering this command from the root of your site installation:

```
composer config repositories.drupal composer https://packages.drupal.org/8
```

2. If you have not already done so, you can optionally override the default location where Composer will put downloaded modules and themes. For example, typically you would want contributed modules to go in a *contrib* subdirectory of the top-level *modules* directory, instead of the default location of *modules*. To make this change, edit the `composer.json` file at the root level in your site installation. Find the line that says `"extra": {`, and add these lines following that line:

```
"installer-paths": {
    "modules/contrib/{$name}": ["type:drupal-module"],
    "modules/custom/{$name}": ["type:drupal-custom-module"],
    "profiles/contrib/{$name}": ["type:drupal-profile"],
    "themes/contrib/{$name}": ["type:drupal-theme"],
```

```
"themes/custom/{$name}": [ "type:drupal-custom-theme" ],  
"libraries/{$name}": [ "type:drupal-library" ],  
"drush/{$name}": [ "type:drupal-drush" ]  
,
```

3. Each time you want to add a contributed module or theme, determine the project's short name. This is the last part of the URL of the project page; for example, the Geofield module, at <https://www.drupal.org/project/geofield>, has short name `geofield`.
4. To download the contributed module or theme, along with its external dependencies, enter the following command at the root of your site (substituting the short name of the module or theme for `geofield`):

```
composer require drupal/geofield
```

Using Composer to update a project's files

Follow these steps to update the files for the core software or a contributed module or theme, after having already started to manage dependencies with Composer:

1. Determine the short name of the project you want to update. For the core software, it is `drupal`. For contributed modules and themes, it is the last part of the URL of the project page; for example, the Geofield module, at <https://www.drupal.org/project/geofield>, has short name `geofield`.
2. Determine how to enter the version number you want to update to. For example, for version 8.x-1.7 of a contributed module, you would enter just the 1.7, and for the core software version 8.3.1, you would enter 3.1.
3. Enter the following command at the root of your site (substituting the short name of the module or theme for `geofield` and the correct version number):

```
composer require drupal/geofield:1.7
```

Additional resources

[Drupal.org community documentation page "Using Composer with Drupal"](#)

Attributions

Adapted by [Jennifer Hodgdon](#) from ["Using Composer with Drupal"](#), copyright 2000-2017 by the individual contributors to the [Drupal Community Documentation](#).

11.10. Deploying New Site Features

Goal

Copy a view that you have created in a local development site to the production site.

Prerequisite knowledge

- [*Bagian 11.8, "Making a Development Site"*](#)
- [*Bagian 11.11, "Synchronizing Configuration Versions"*](#)
- [*Bagian 11.7, "Concept: Development Sites"*](#)

Site prerequisites

- The core Configuration Manager module must be installed in both the development and production sites. See [*Bagian 4.3, "Installing a Module"*](#) for instructions on installing core modules.
- The Vendor content type must exist in both the development and production sites, with the same fields. See [*Bagian 6.1, "Adding a Content Type"*](#).
- The Vendors view must exist in the development site but not the production site. See [*Bagian 9.3, "Creating a Content List View"*](#).

Steps

1. Open the local development site.
2. In the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Export > Single item* (`admin/config/development/configuration/single/export`). The *Single export* page appears.
3. Select *View* from the *Configuration type* list.
4. Select *Vendors* from the *Configuration name* list. The configuration appears in the textarea.
5. Copy the configuration from the textarea.

The screenshot shows the 'Single export' page. At the top, there are tabs for 'Synchronize', 'Import', and 'Export'. Below them, two options are shown: 'Full archive' and 'Single item', with 'Single item' being selected. A breadcrumb navigation path is visible: Home > Administration > Configuration > Development > Synchronize. A message below the path says 'Choose a configuration item to display its YAML structure.' Under 'Configuration type', 'View' is selected. In the 'Configuration name' field, 'Vendors (vendors)' is entered. A large text area titled 'Here is your configuration:' contains the following YAML code:

```
uuid: bf3b16f1-4807-4899-9c76-94dd2f97b3c7
langcode: en
status: true
dependencies:
  config:
    - core.entity_view_mode.node.teaser
    - field.storage.node.body
    - field.storage.node.field_main_image
    - image.style.medium
    - node.type.vendor
    - system.menu.main
  module:
    - image
    - node
    - text
    - user
id: vendors
label: Vendors
module: views
description: ''
tag: ''
base_table: node_field_data
base_field: nid
core: 8.x
```

6. Open the production site.
7. In the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization*. The *Import* page appears.
8. Select *View* from the *Configuration type* list.
9. Paste the configuration in the text area.
10. Click *Import*. The confirmation page appears.
11. Click *Confirm*.
12. Verify that the view was imported by navigating in the *Manage* administrative menu to *Structure > Views*.

Expand your understanding

The steps in this topic show how to export and import a single configuration item. However, often if you develop functionality on a development website and want to transfer it to your production site, you will need to transfer multiple configuration items. For instance, if you developed a new content type with fields, you would need to transfer several configuration items for each field, one for the content type itself, and possibly multiple view mode and form mode items, and they would have to be transferred in the right order. Getting this right can be both tedious and difficult.

As an alternative, you can export and import the complete configuration of the site. For this, you would need a local development site that is a clone of the production site (see [Bagian 11.8, "Making a Development Site"](#)), and then you can follow the steps in [Bagian 11.11, "Synchronizing Configuration Versions"](#) to synchronize configuration between development and production sites.

Another alternative is to use the [contributed Features module](#), which allows exporting and importing bundled functionality (for example, all the configuration for a photo gallery).

Finally, if you do not see the effect of these changes in your site, you might need to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

Related concepts

- [Bagian 11.7, "Concept: Development Sites"](#)
- [Bagian 2.6, "Concept: Editorial Workflow"](#)

Attributions

Written by [Boris Doesborg](#).

11.11. Synchronizing Configuration Versions

Goal

Synchronize the configuration between a development and live site.

Prerequisite knowledge

- [Bagian 1.5, "Concept: Types of Data"](#)
- [Bagian 3.3, "Preparing to Install"](#)
- [Bagian 11.7, "Concept: Development Sites"](#)
- [Bagian 11.8, "Making a Development Site"](#)

Site prerequisites

- You must have a development copy of your production site. See [Bagian 11.8, "Making a Development Site"](#).
- The core Configuration Manager module must be installed on both the development and production sites. See [Bagian 4.3, "Installing a Module"](#) for instructions on how to install core modules.
- You must have changed configuration on either the production or development site (the *source site*), and want to synchronize the changes to the other site (the *destination site*). As an example, you can develop a new content type, fields, and views on your development site, and when it is all working correctly, deploy the changes to the live site.
- All configuration that should not be synchronized between the source and destination sites must be stored in configuration overrides in the *settings.php* file rather than in the database. See [Bagian 11.8, "Making a Development Site"](#).

Steps

1. In the source site, in the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Export* (*admin/config/development/configuration/full/export*).
2. Click *Export*. Your site will generate an archive of the full site configuration. Save the file on your local computer.
3. In the destination site, in the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Import* (*admin/config/development/configuration/full/import*).
4. Browse to find the downloaded configuration archive, and click *Upload*. Your configuration archive will be uploaded to the destination site, and you will be redirected to the configuration *Synchronize* page (*admin/config/development/configuration*) with a message saying your files were uploaded.
5. Verify that the differences shown on the page are what you expect. You may see configuration items that have been added, deleted, or changed; for changed items, you can click *View differences* to see what the changes are.
6. When you are satisfied, click *Import all* to import the configuration changes.

Expand your understanding

- If the changes you have made involve only one configuration item (such as one view), you can use the single configuration export/import feature to deploy the change between sites. See [Bagian 11.10, "Deploying New Site Features"](#).
- After the step where you export the full configuration from the source site, you might also want to unpack the archive and commit it to a version control system, such as Git, to track changes in your site configuration. See [Bagian 3.2, "Concept: Additional Tools"](#).

Attributions

Written by [Jennifer Hodgdon](#).

Bab 12. Mencegah dan Memperbaiki Masalah

12.1. Concept: Cache

Prerequisite knowledge

[Bagian 1.1, "Concept: Drupal as a Content Management System"](#)

What is the page cache?

The software that runs your site, on each page request, must perform calculations and retrieve data from the database, in order to compose the page that is sent to the web browser or other application that is accessing the site. These calculations take time, which can mean that your page load time is longer than would be desirable.

There are several ways that page load time can be sped up, including installing software on the server. The system includes the core Internal Page Cache and Dynamic Page Cache modules, which do not require any additional server software; they use a *database cache* mechanism to speed up your site. The way these modules work is that during page calculations, intermediate results and the final page output are stored in a special database area (known as the *cache*). Then the next time a compatible request is made, intermediate or final results, as appropriate, can be retrieved and used rather than redoing the entire calculation. In addition, when content or data that affects a particular calculation is updated, the affected cached data is removed from the cache, forcing that part of the calculation to be redone the next time it is needed.

These caching modules normally work reasonably well, and offer at least some speed-up for most sites. However, sometimes the page cache can have problems, such as:

- Corrupted data in the cache, leading to garbled or incorrect page output
- Old data remaining in the cache too long, leading to outdated page output
- Insufficient caching, leading to slow page loads

What other data is cached?

Independent of whether the two page cache modules are installed on your site, the software that your site runs will still cache the output of many internal calculations. The core systems that cache data include:

- The theme system caches information in the database cache about which template files are used to render various types of data. If you are developing a new theme and add a new template file, you'll need to clear this cache to have your theme file recognized.

- CSS and JavaScript files can optionally be optimized and compressed (depending on your site settings). If so, the compressed versions are stored in the file system so that they don't have to be re-optimized too often. If you are developing a module or theme, you may need to either turn off or clear this file cache to have changes to CSS or JavaScript files be recognized.
- The system locates certain low-level PHP functions and classes, such as *hook implementations* and *plugin classes*, from your installed modules and stores information about which module has which functionality. If you are developing a new module or adding features to an existing module, you may need to clear this cache to have your new features be recognized.

Related topics

If you have problems with your site, the first thing to try to fix it is usually to clear the cache. See [Bagian 12.2, "Clearing the Cache"](#) for more information.

Additional resources

Learn about additional caching and performance optimization methods in the [Drupal.org community documentation page "Caching to improve performance"](#).

Attributions

Written by [Jennifer Hodgdon](#).

12.2. Clearing the Cache

Goal

Clear or rebuild your site's internal caches to ensure they are up-to-date with the most recent data, using the user interface or Drush.

Prerequisite knowledge

- [Bagian 12.1, "Concept: Cache"](#)
- [Bagian 3.2, "Concept: Additional Tools"](#)

Site prerequisites

If you want to use Drush to clear the cache, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

You can use the administrative interface or Drush to clear the cache.

Using the administrative interface

1. In the *Manage* administrative menu, navigate to *Configuration > Development > Performance* (*admin/config/development/performance*).
2. Click *Clear all caches*.
3. A message saying the cache has been cleared appears at the top of the page.

Using Drush

You can use one of two commands:

- Use the command `drush cache-rebuild` to clear and rebuild all cached data for a site. After running this command, you will see the output message "Cache rebuild complete."
- Use the command `drush cc` to see a list of individual caches and then choose the specific cache you would like to clear. Running this command should produce output like the following:

```
> drush cc
Enter a number to choose which cache to clear.
[0] : Cancel
[1] : drush
[2] : theme-registry
[3] : menu
[4] : css-js
[5] : block
[6] : module-list
[7] : theme-list
[8] : render
[9] : views
```

Choose a cache to clear by entering the number associated with that cache. Press "Enter" to continue.

Additional resources

[Drupal.org community documentation page "Clearing or rebuilding Drupal's cache"](#)

Attributions

Adapted and edited by [Joe Shindelar](#) and [Jack Haas](#) from "[Clearing or rebuilding Drupal's cache](#)", copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).

12.3. Concept: Data Backups

Prerequisite knowledge

- [Bagian 1.1, "Concept: Drupal as a Content Management System"](#)

- [Bagian 3.3, "Preparing to Install"](#)

What is a site backup?

If something happens to the computer (or computers) that your site and its database are running on, or if you lose access to this computer, you could lose some or all of your site's data. You could also lose data if your site is hacked, or if someone with administrative privileges on your site deletes or alters data mistakenly from the administrative interface. In order to prevent scenarios like this from being permanent, expensive data losses, it is important to make regular backups of your site's data, and to store them in a location that is separate from the computer where your site is running.

The frequency with which you should make data backups, and how many backups you should keep, depends on how frequently your site is changing. If you have a very large amount of content on your site that is being added to or updated many times per day, you would want to make more frequent backups than you would for a site that changes rarely. Also consider that some time could pass between when a data problem occurs and when you notice that it is a problem, so storing a sequence of backups (so that you can go back to the last known good data and retrieve that), rather than overwriting a single backup repeatedly, is a good practice.

Another consideration is that whatever format you store your backups in, it is a good idea to verify that you can actually retrieve lost data from your backups. You might want to test several possible data loss scenarios, and make sure that your data can be restored to the site in all cases.

In order to make a complete backup of your site, you will need to make copies of the following:

- The data in the *sites* directory, including the *sites/default/settings.php* file.
- The data in your site's database. A few tables can be truncated, such as those storing the temporary data cache and user login session information, but it is always safe to back up the entire database.
- Uploaded files, such as images and other attachments. The location of these files is configurable; the standard location is the *sites/default/files* directory under your site root. In the *Manage* administrative menu, navigate to *Configuration > Media > File system (admin/config/media/file-system)* to check the file upload locations; to change them, you'll need to edit your *settings.php* file.
- Modules, themes and any other software files you have customized. You can find customized modules and themes in the *modules* and *themes* directories respectively. Some people prefer to back up all software files, including core files and contributed modules and themes (which you could recover by downloading them again from the source), rather than trying to pick out specific files that definitely need to be backed up.

You can perform a test to confirm whether your backup has been done right by making a development copy of the site (see [Bagian 11.8, "Making a Development Site"](#)).

Related topics

- [Bagian 13.5, "Updating the Core Software"](#)
- [Bagian 11.8, "Making a Development Site"](#)
- [Bagian 12.1, "Concept: Cache"](#)

Additional resources

- [Drupal.org community documentation page "Backing up a site"](#)
- The [contributed Backup and Migrate module](#), which can be used to set up automatic backups of the database and uploaded files.

Attributions

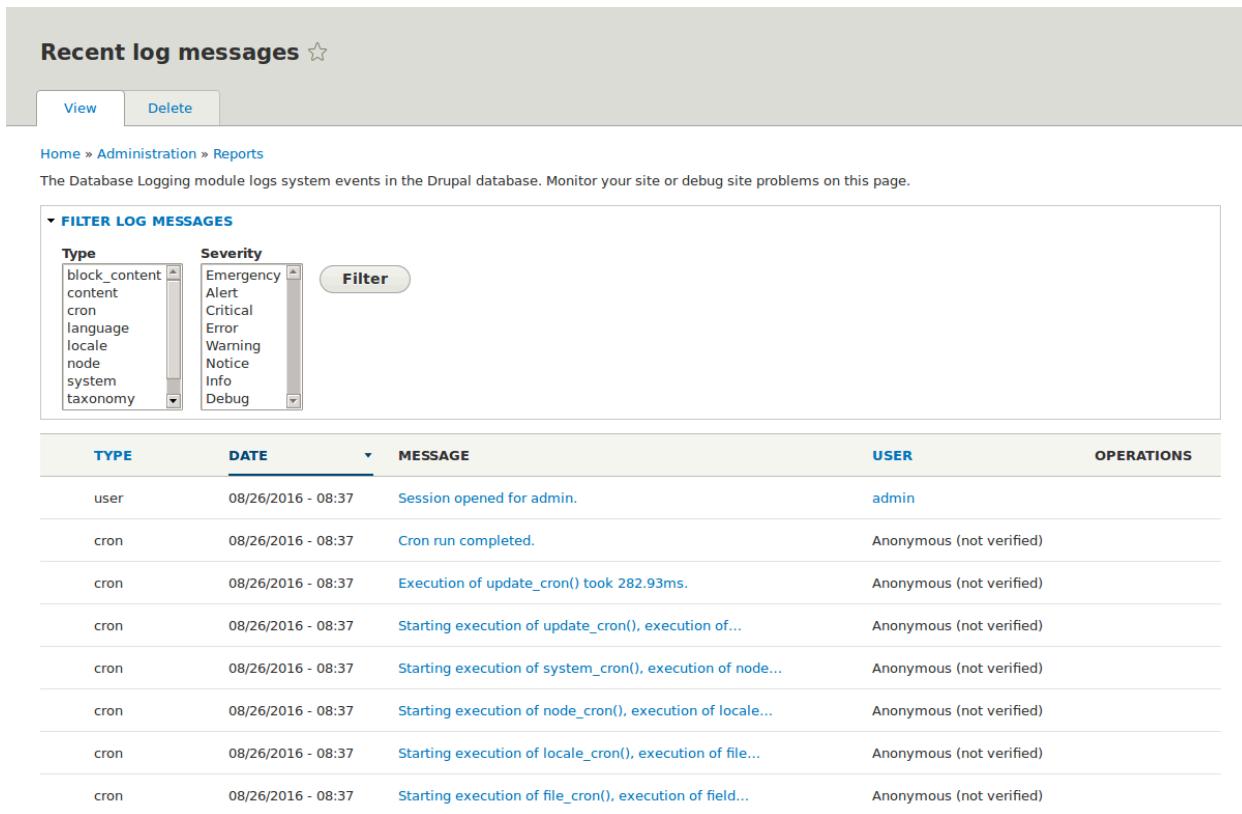
Written by [Jennifer Hodgdon](#).

12.4. Concept: Log

What is a Log?

Your site captures system events in a log to be reviewed by an authorized individual at a later time. The log is a list of recorded events containing usage data, performance data, errors, warnings, and operational information. It is vital to check the log on a regular basis as it is often the only way to tell what is going on.

You can find your site's recent log messages in the *Manage* administrative menu by navigating to *Reports > Recent log messages* ([admin/reports/dblog](#)).



Recent log messages ☆

[View](#) [Delete](#)

Home » Administration » Reports

The Database Logging module logs system events in the Drupal database. Monitor your site or debug site problems on this page.

FILTER LOG MESSAGES

Type	Severity
block_content	Emergency
content	Alert
cron	Critical
language	Error
locale	Warning
node	Notice
system	Info
taxonomy	Debug

Filter

TYPE	DATE	MESSAGE	USER	OPERATIONS
user	08/26/2016 - 08:37	Session opened for admin.	admin	
cron	08/26/2016 - 08:37	Cron run completed.	Anonymous (not verified)	
cron	08/26/2016 - 08:37	Execution of update_cron() took 282.93ms.	Anonymous (not verified)	
cron	08/26/2016 - 08:37	Starting execution of update_cron(), execution of...	Anonymous (not verified)	
cron	08/26/2016 - 08:37	Starting execution of system_cron(), execution of node...	Anonymous (not verified)	
cron	08/26/2016 - 08:37	Starting execution of node_cron(), execution of locale...	Anonymous (not verified)	
cron	08/26/2016 - 08:37	Starting execution of locale_cron(), execution of file...	Anonymous (not verified)	
cron	08/26/2016 - 08:37	Starting execution of file_cron(), execution of field...	Anonymous (not verified)	

Attributions

Adapted by [Diána Lakatos](#) from "[Reports](#)" copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#)

12.5. Concept: Status Report

What is a Status Report?

The status report is a short overview of your site's parameters as well as any problems detected with your installation. It may be useful to copy and paste this information into support requests filed on [Drupal.org](#)'s support forums and project issue queues or when asking for help on other channels.

You can find the status report in the *Manage* administrative menu by navigating to *Reports > Status report (admin/reports/status)*.

Status report ☆

[Home](#) » [Administration](#) » [Reports](#)

Here you can find a short overview of your site's parameters as well as any problems detected with your installation. It may be useful to copy and paste this information into support requests filed on [Drupal.org](#)'s support forums and project issue queues. Before filing a support request, ensure that your web server meets the [system requirements](#).

Drupal	8.2.0-dev
Access to update.php	Protected
Configuration files	Protected
Cron maintenance tasks	Last run 1 minute 5 seconds ago You can run cron manually . To run cron from outside the site, go to example.com/cron /V_A2qVbam3F_hvAxbBG0hHuOzorCEF3fRbi43bWdecHN0YY9RZ_Yy5aUC_1nsiHoKb_HRjOlg?absolute=1
Database system	MySQL, MariaDB, Percona Server, or equivalent
Database system version	5.5.47-Ubuntu0.14.04.1
✖ Database updates	Out of date Some modules have database schema updates to install. You should run the database update script immediately.
⚠ Drupal core update status	Unknown release date (version 8.1.8 available) There was a problem checking available updates for Drupal. See the available updates page for more information and to install your missing updates.
Entity/field definitions	Up to date
File system	Writable (<i>public</i> download method)
GD library	2.1.1-dev
GD library PNG support	2.1.1-dev

Related topics

[Bagian 14.2, "Getting Support"](#)

Attributions

Adapted by [Diána Lakatos](#) from ["Reports"](#) copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#)

Bab 13. Keamanan dan Pemeliharaan

13.1. Concept: Cron

What are cron tasks?

To ensure that your site and its modules continue to function well, a group of administrative operations should be run periodically. These operations are called *cron* tasks. Examples of cron tasks are: checking for module and theme updates, indexing content for search, or cleaning up temporary files.

What is the relationship between the site's cron tasks and Unix cron?

Linux/Unix-based operating systems have a cron scheduler that can be used to run periodic tasks. You can use the server's cron scheduler to schedule runs of the site's cron tasks. Alternatively, you can use the core Automated Cron module to run tasks. You can check the site's cron tasks' status in the status report.

Related topics

- [*Bagian 13.2, "Configuring Cron Maintenance Tasks"*](#)
- [*Bagian 12.5, "Concept: Status Report"*](#)

Additional resources

[Drupal.org community documentation page "Setting up cron"](#)

Attributions

Written and edited by [Diána Lakatos](#) at [Pronovix](#), [Dave Hansen-Lange](#) at [Advomatic](#), and [Boris Doesborg](#).

13.2. Configuring Cron Maintenance Tasks

Goal

Check whether cron maintenance tasks are run regularly, and if not, either install the core Automated Cron module or run cron maintenance tasks from outside the website.

Prerequisite knowledge

[Bagian 13.1, "Concept: Cron"](#)

Steps

1. Review the *Status report* (see [Bagian 12.5, "Concept: Status Report"](#)) to see when cron maintenance tasks were last run.

If you installed the website using the core Standard installation profile (or similar), then cron maintenance tasks might already be running via the core Automated Cron module. By default, these tasks are run about every three hours.

2. Choose whether to run cron maintenance tasks using the core Automated Cron module, or by other means. The core Automated Cron module might not be suitable for some websites because:
 - Each time someone accesses a page on the site, the module checks how long it has been since cron maintenance tasks have last run, and then runs them if necessary. If no one visits the website for a long time, cron maintenance tasks will not be run.
 - Cron maintenance tasks are run after the page has been generated. This means there is less time for the tasks to be run before various server timeouts are reached (for example, PHP execution timeout). If this happens, the logs (see [Bagian 12.4, "Concept: Log"](#)) will show error messages that cron is unable to complete.
 - There is a small *scalability* cost associated with the core Automated Cron module. This is because one of the web server's processes is occupied (and can't serve other web pages) until the cron maintenance tasks are complete.
3. If you want to use the core Automated Cron module, first make sure it is installed (it is installed with the core Standard install profile; see [Bagian 4.3, "Installing a Module"](#) if it is not installed).

Next, configure the module to control how frequently cron maintenance tasks are run. In the *Manage* administrative menu, navigate to *Configuration > System > Cron* (`admin/config/system/cron`). Choose the desired interval in the *Cron settings* section, and click *Save configuration*.

The screenshot shows the 'Cron' administration page. At the top, it says 'Cron' with a star icon. Below that is a breadcrumb trail: Home > Administration > Configuration > System. A note states: 'Cron takes care of running periodic tasks like checking for updates and indexing content for search.' A 'Run cron' button is available. Below it, a message says 'Last run: 2 minutes 28 seconds ago.' Another button 'Run cron' is shown with the URL 'http://example.com/cron/V_A2qVbam3F_hvAXbBG0hHuOzorCEf3fHRbi43bWdecHN0YY9RZ_Yy5aUC_1nsiHoKb_HRjOlg'. A 'CRON SETTINGS' section is expanded, showing 'Run cron every 3 hours'. A link to 'More information about setting up scheduled tasks' is provided. A 'Save configuration' button is at the bottom.

4. If you want to run cron maintenance tasks from outside the website, uninstall the core Automated Cron module (see [Bagian 4.4, "Uninstalling Unused Modules"](#)). Next, find the cron URL. This URL is shown in the *Status report* (see [Bagian 12.5, "Concept: Status Report"](#)), and in the *Cron* administration page (see previous step). The URL looks like this: <http://www.example.com/cron/0MgWtfB33FYbbQ5UAC3L0LL3RC0PT3RNUBZILLA0Nf1Re>

Whenever this URL is visited, cron maintenance tasks will run. Set up one of the following schedulers to access this URL regularly:

- [The Cron daemon](#) (Linux, OS X, Solaris, BSD)
- [Scheduled Tasks](#) (Windows)
- A cron SASS provider (software as a service)
- A cron manager provided by your web hosting provider (see the documentation provided by your provider)

Related concepts

[Bagian 13.3, "Concept: Security and Regular Updates"](#)

Additional resources

- [Drush page "Running Drupal cron tasks from Drush"](#)
- [Drupal.org community documentation page "Setting up cron"](#)

Attributions

Written and edited by [Dave Hansen-Lange](#) at [Advomatic](#), [Boris Doesborg](#), and [Jennifer Hodgdon](#).

13.3. Concept: Security and Regular Updates

Prerequisite knowledge

- [Bagian 1.1, "Concept: Drupal as a Content Management System"](#)
- [Bagian 1.2, "Concept: Modules"](#)
- [Bagian 1.3, "Concept: Themes"](#)
- [Bagian 1.6, "Concept: The Drupal Project"](#)

What are security updates?

Any software occasionally has bugs, and sometimes these bugs have security implications. When security bugs are fixed in the core software, modules, or themes that your site uses, they are released in a *security update*. You will need to apply security updates in order to keep your site secure. See [Bagian 13.4, "Keeping Track of Updates"](#) to learn how to be notified of security updates by email, and [Bagian 13.5, "Updating the Core Software"](#), [Bagian 13.6, "Updating a Module"](#), and [Bagian 13.7, "Updating a Theme"](#) to learn how to make updates.

What is the security team?

The Drupal open-source project has a team of volunteers who track security-related bugs and release security updates. They also help other developers fix bugs, and maintain information for users on how to keep their websites secure. You can learn more about the security team and their practices and processes at the [Drupal.org Security Team](#) page.

How are security bugs reported?

It is important that security problems be kept confidential until they are fixed, so that sites are less likely to be compromised before they can be secured. If you find a potential security problem in any of the software you downloaded from the [Drupal.org](#) website, follow the procedures on the [Drupal.org Security Team](#) page to report it.

What are regular (non-security) updates?

The core software, modules, and themes also periodically have regular updates to add new features and fix bugs. These updates are less critical than security updates. As a general best practice, updates should be applied as long as they do not cause problems with your site. Testing on a development copy of your site is always a good idea before applying updates on a live site. This is because some updates may include changes that are not compatible with the modules or themes on your site, or that will break a particular functionality on your site.

Related topics

- [Bagian 13.4, "Keeping Track of Updates"](#)
- [Bagian 13.5, "Updating the Core Software"](#)

- [Bagian 13.6, "Updating a Module"](#)
- [Bagian 13.7, "Updating a Theme"](#)
- [Bagian 11.8, "Making a Development Site"](#)

Additional resources

- [Drupal.org community documentation page "Securing your site"](#)
- ["Security advisories" on Drupal.org](#)
- [Drupal.org Security team](#)

Attributions

Written and edited by [Boris Doesborg](#) and [Jennifer Hodgdon](#).

13.4. Keeping Track of Updates

Goal

Keep your site safe and up-to-date by keeping up with the latest security updates.

Prerequisite knowledge

[Bagian 13.3, "Concept: Security and Regular Updates"](#)

Steps

There are a few different ways to receive notifications of security releases:

- In the *Manage* administrative menu, navigate to *Reports > Available updates > Settings*. Enter the email addresses to be notified when updates are available. You can also specify whether you want daily or weekly updates. Click *Save configuration*.
- Subscribe to the security announcements email list. To subscribe, log in to *Drupal.org*, go to your user profile page and subscribe to the security newsletter on the *Edit > My newsletters* tab.
- Follow @drupalsecurity on Twitter.
- Subscribe to RSS feeds for [core security updates](#), [contributed project updates](#) and [public service announcements](#).

Related concepts

- [Bagian 13.3, "Concept: Security and Regular Updates"](#)
- [Bagian 13.5, "Updating the Core Software"](#)
- [Bagian 13.6, "Updating a Module"](#)

- [Bagian 13.7, "Updating a Theme"](#)

Additional resources

- ["Security advisories" on Drupal.org](#)
- [Drupal.org Security Team](#)
- [@drupalsecurity on Twitter](#)

Attributions

Written by [Sarah German](#) at [Advomatic](#).

13.5. Updating the Core Software

Goal

Update the core software, either through the administrative interface or by using Drush.

Site prerequisites

- If you want to use Drush, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).
- If your site is live, you should test this process in a development environment before running it on your production site. See [Bagian 11.8, "Making a Development Site"](#).

Steps

1. Make a complete backup of your site. Refer to [Bagian 12.3, "Concept: Data Backups"](#).
2. Open `settings.php` (`/sites/default/settings.php`) in any plain text editor. Find the line with the `$settings[update_free_access]` variable. By default, it is set to "FALSE" due to security reasons. Change the setting to "TRUE":

```
$settings['update_free_access'] = TRUE;
```
3. Disable any caching technique (memcache, varnish, and so on) your application might be using.
4. Put your site in maintenance mode. See [Bagian 11.2, "Enabling and Disabling Maintenance Mode"](#).
5. If you are using Composer to manage dependencies, or using Git, skip the next six steps, and instead see [Bagian 11.9, "Using Composer and Git to Download Files"](#) for instructions on downloading updated files. Continue with the `update.php` step.
6. Download the tar.gz or zip file archive for the latest version of Drupal 8.x core from [Drupal.org Drupal Core Downloads](#). See [Bagian 3.3, "Preparing to Install"](#) for more details on how to find the latest version.
7. Upload the archive file to your web hosting server.

8. Extract the archive file to a temporary directory on your server (should be outside the directory where the site is hosted). Your hosting control panel's file manager should provide a way to extract the files. Or, if you have terminal access to your hosting server (running Linux), you can use a command like:

```
tar -xzf drupal-8.3.2.tar.gz
```

9. You can also use Drush to download the archive and extract it in one step, by navigating to the temporary directory on the server and entering this command:

```
drush dl drupal
```

10. In your site hosting directory, delete the *core* and *vendor* directories, and all files that are not in a subdirectory, including *.htaccess*, *composer.json*, and *autoload.php*. Don't delete custom and customized files because you may end up losing the custom functionality stored in them.

11. Copy the *core* and *vendor* directories and the non-custom/non-customized files that you deleted in the preceding step from the temporary directory to your site directory.

12. Run the *update.php* script using either of the following:

- Visit <http://www.example.com/update.php> in your browser (where *www.example.com* is your site's URL). Click *Continue* in the first screen to run the updates and successfully complete the script.
- Run the following Drush command: `drush updb`

13. If you get any error or warning, re-run the *update.php* script again till all the updates have been completed successfully.

14. Open *settings.php* (*/sites/default/settings.php*) in a text editor. Find the line with the *\$settings[update_free_access]* variable and update it to "FALSE":

```
$settings['update_free_access'] = FALSE;
```

15. Click *Administration pages* to return to the administration section of your site.

16. Take your site out of maintenance mode. See [Bagian 11.2, "Enabling and Disabling Maintenance Mode"](#).

17. Clear the cache. See [Bagian 12.2, "Clearing the Cache"](#).

18. Re-enable any caching technique you disabled at Step 3.

19. You should have the updated version running. You can verify the current version of your software by checking the *Status report* (see [Bagian 12.5, "Concept: Status Report"](#)).

Expand your understanding

- [Bagian 3.2, "Concept: Additional Tools"](#)
- [Bagian 11.8, "Making a Development Site"](#)
- [Bagian 12.3, "Concept: Data Backups"](#)

Related concepts

- [Bagian 12.5, "Concept: Status Report"](#)

Additional resources

- ["Drupal Core Downloads" page on Drupal.org](#)
- ["Registry Rebuild" page on Drupal.org](#)
- The file /core/UPDATE.txt within your installation.

Attributions

Written and edited by [Surendra Mohan](#), [Boris Doesborgh](#), and [Jojo Alphonso](#) at [Red Crackle](#).

13.6. Updating a Module

Goal

Update a contributed module and run the *Database updates* script.

Prerequisite knowledge

- [Bagian 13.3, "Concept: Security and Regular Updates"](#)
- [Bagian 13.1, "Concept: Cron"](#)

Site prerequisites

- A contributed module has been installed and there is an update available for it. See [Bagian 11.3, "Downloading and Installing a Module from Drupal.org"](#) and [Bagian 13.4, "Keeping Track of Updates"](#).
- If your site is live, you should test this process in a development environment before running it on your production site. See [Bagian 11.8, "Making a Development Site"](#).
- You have created a full-site backup. See [Bagian 12.3, "Concept: Data Backups"](#).
- If you want to use the user interface, the core Update Manager module must be installed. See [Bagian 4.3, "Installing a Module"](#) for instructions on installing core modules.
- If you want to use Drush, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

Before you start, check for module-specific update instructions. This is typically necessary while updating modules that involve the usage of third-party libraries. Read and understand all module-specific requirements before proceeding with the updates. To find instructions, check the module's project page *Read Documentation* link.

To view further instructions, download the tar.gz or .zip file from the project page to your local computer. Unzip the file and look for *README.txt*, *INSTALL.txt*, and *UPGRADE.txt* that come with the module's installation file. Also, review the release notes on the project page by clicking the version number you're downloading.

Downloads

Version	Download	Date
8.x-1.19	tar.gz (17.07 KB) zip (27.02 KB)	2017-Apr-06
Development releases		
8.x-1.x-dev	tar.gz (19.49 KB) zip (29.81 KB)	2017-Jun-11

[View all releases](#)

You can use the administrative interface or Drush to update a contributed module. If you are updating a custom module rather than a contributed module, if you see a message saying *Installing modules and themes requires FTP access to your server*, or if the steps below do not work to obtain the new module files, follow the steps in [Bagian 11.6, "Manually Downloading Module or Theme Files"](#). You can then continue here with step 6 in the instructions for the administrative interface below.

Using the administrative interface

1. Put your site in maintenance mode. See [Bagian 11.2, "Enabling and Disabling Maintenance Mode"](#).
2. In the *Manage* administrative menu, navigate to *Reports > Available updates > Update* ([admin/reports/updates/update](#)).
3. Find and check the module in the list. Click *Download these updates* for the module.

The screenshot shows the 'Available updates' page. At the top, there is a navigation bar with tabs: 'List' (selected), 'Update', and 'Settings'. Below the navigation bar, the breadcrumb trail shows: Home » Administration » Reports » Available updates. A message indicates 'Last checked: 4 minutes 5 seconds ago (Check manually)'. A table lists available updates, showing the module name, installed version, and recommended version. One row for 'Admin Toolbar' is shown, with the installed version being '8.x-1.15' and the recommended version being '8.x-1.16 (Release notes)'. A large button labeled 'Download these updates' is visible. Below this, a section titled 'Manual updates required' lists 'Updates of Drupal core are not supported at this time.' Another table shows the status for 'Drupal core', with the installed version being '8.2.0-dev' and the recommended version being '8.1.8 (Release notes)'.

NAME	INSTALLED VERSION	RECOMMENDED VERSION
Admin Toolbar	8.x-1.15	8.x-1.16 (Release notes)

NAME	INSTALLED VERSION	RECOMMENDED VERSION
Drupal core	8.2.0-dev	8.1.8 (Release notes)

4. Click *Continue*.
5. Click *Run database updates*. If you obtained the new module files manually, start with this step, and reach the database updates page by typing the URL `example.com/update.php` in your browser.
6. Click *Continue* and apply all updates. The database update scripts will be executed.
7. Click *Administration pages* to return to the administration section of your site.
8. Take your site out of maintenance mode. See [*Bagian 11.2, "Enabling and Disabling Maintenance Mode"*](#).
9. Clear the cache (refer to [*Bagian 12.2, "Clearing the Cache"*](#)).

Using Drush

1. Find the project name for the module you wish to update. It is the last segment of the module's project page URL. For example, if the project URL is `https://www.drupal.org/project/admin_toolbar`, the project name is "admin_toolbar".
2. Run the following Drush command, giving the project name (for example, `admin_toolbar`) as a parameter (if you have more than one module to update, add the additional module project names to the end of the command, separated by spaces):

```
drush up admin_toolbar
```

3. Follow the instructions on the screen.

Expand your understanding

- Review the site log (refer to [*Bagian 12.4, "Concept: Log"*](#)) once the updates are complete to check for errors.
- [*Bagian 13.7, "Updating a Theme"*](#)

Additional resources

[*Drupal.org community documentation page "Updating modules"*](#)

Attributions

Adapted by [*Boris Doesborgh*](#), and [*Sarah German*](#) at [*Advomatic*](#), from [*"Updating modules"*](#), copyright 2000-2016 by the individual contributors to the [*Drupal Community Documentation*](#).

13.7. Updating a Theme

Goal

Update a contributed theme on your site and run the *Database Updates* script.

Prerequisite knowledge

- [Bagian 13.3, "Concept: Security and Regular Updates"](#)
- [Bagian 13.1, "Concept: Cron"](#)

Site prerequisites

- A contributed theme has been installed and there is an update available for it. See [Bagian 11.5, "Downloading and Installing a Theme from Drupal.org"](#) and [Bagian 13.4, "Keeping Track of Updates"](#).
- If your site is live, you should test this process in a development environment before running it on your production site. See [Bagian 11.8, "Making a Development Site"](#).
- You have created a full site backup. See [Bagian 12.3, "Concept: Data Backups"](#).
- If you want to use the user interface, the core Update Manager module must be installed. See [Bagian 4.3, "Installing a Module"](#) for instructions on installing core modules.
- If you want to use Drush, Drush must be installed. See [Bagian 3.2, "Concept: Additional Tools"](#).

Steps

You can use the administrative interface or Drush to update a contributed theme. If you are updating a custom theme rather than a contributed theme, if you see a message saying *Installing modules and themes requires FTP access to your server*, or if the steps below do not work to obtain the new theme files, follow the steps in [Bagian 11.6, "Manually Downloading Module or Theme Files"](#), and then continue with step 6 in the instructions for administrative interface below.

Using the administrative interface

1. Put your site in maintenance mode. See [Bagian 11.2, "Enabling and Disabling Maintenance Mode"](#).
2. In the *Manage* administrative menu, navigate to *Reports > Available updates > Update* ([admin/reports/updates/update](#)).
3. Find and check the theme in the list. Click *Download these updates* for the theme.

The screenshot shows the 'Update' page with the title 'Update ★'. There are three tabs: 'List', 'Update' (which is selected), and 'Settings'. Below the tabs, the breadcrumb navigation shows: Home > Administration > Reports > Available updates. It indicates that the last check was 1 minute 1 second ago, with a link to 'Check manually'. A table lists updates for the 'MAYO (Theme)' and 'Drupal core'. For the theme, the installed version is 8.x-1.2 and the recommended version is 8.x-1.3 (with a link to 'Release notes'). For Drupal core, the installed version is 8.2.0-dev and the recommended version is 8.1.8 (with a link to 'Release notes'). A button labeled 'Download these updates' is visible.

<input type="checkbox"/> NAME	INSTALLED VERSION	RECOMMENDED VERSION
<input type="checkbox"/> MAYO (Theme)	8.x-1.2	8.x-1.3 (Release notes)

NAME	INSTALLED VERSION	RECOMMENDED VERSION
Drupal core	8.2.0-dev	8.1.8 (Release notes)

4. Click *Continue*.
5. Click *Run database updates*. If you obtained the new theme files manually, start with this step, and reach the database updates page by typing the URL `example.com/update.php` in your browser.
6. Click *Continue* to run the updates. The database update scripts will be executed.
7. Click *Administration pages* to return to the administration section of your site.
8. Take your site out of maintenance mode. See [Bagian 11.2, "Enabling and Disabling Maintenance Mode"](#).
9. Clear the Drupal cache (refer to [Bagian 12.2, "Clearing the Cache"](#)).

Using Drush

1. Find the project name for the theme you wish to update, which is the last segment of the theme's project page URL. For example, if the project URL is `https://www.drupal.org/project/mayo`, the project name is "mayo".
2. Run the following Drush command, giving the project name (for example, `mayo`) as a parameter:

```
drush up mayo
```

3. Follow the instructions on the screen.

Expand your understanding

- Review the site log, see [Bagian 12.4, "Concept: Log"](#), once the updates are complete to check for errors.

- *Bagian 13.6, "Updating a Module"*

Attributions

Written by [*Boris Doesborg*](#).

Bab 14. Pemikiran Akhir

14.1. Connecting with the Community

Prerequisite knowledge

[Bagian 1.6, "Concept: The Drupal Project"](#)

How can you connect with the community?

The Drupal project has a world-wide community of developers and users. One of the best ways to improve your knowledge of the platform is to connect with others that are using it, and get involved in the open-source community. There are many ways that you can get started:

Attend an event

There are both regional and international Drupal events held around the world. See the ["DrupalCon" page on Drupal.org](#) to find international events, and the ["Event Calendar" on groups.drupal.org](#) or [drupical.com](#) to find regional events.

Join a local group

There are Drupal user groups all around the world. Many of them have regular meetings, which you can attend to learn more about Drupal and connect to other Drupal users. Find local user groups on [groups.drupal.org](#).

Participate in a topical or language group

There are also interest groups for a wide range of topics, which have on-line discussion forums. Find topical groups on [groups.drupal.org](#). Many languages have their own websites too; you can find them on the ["Language-specific communities" page on Drupal.org](#).

Chat online

The Drupal project uses IRC for on-line chatting. There are regional, topical, and general-purpose chat groups available. Find out more on the ["IRC" page on Drupal.org](#).

Report a problem

See below.

Contribute

You can contribute your time and expertise to the community in many ways, such as:

- Developing modules or themes. See the ["Contribute to Development" page on Drupal.org](#), or improving them (using the issue queues for existing projects).
- Translating the user interface. See [localize.drupal.org](#).
- Writing documentation. See ["Contribute to Documentation" on Drupal.org](#).

- Answer Support questions. See [*Bagian 14.2, "Getting Support".*](#)

How can you report a problem or suggest a feature?

Each project within the community (such as the Drupal Core project for the base software, and projects for each contributed theme and module) uses *issues* to keep track of software bugs and plans for new features. You can participate by creating a bug report when you find a problem, creating a feature request, or commenting on existing issues. Search before creating an issue, to make sure that the problem or feature has not already been reported or requested. See the [*Drupal.org page "Use the issue queue"*](#) and the [*Drupal.org page "Reporting a problem"*](#) for more information.

If you find a problem that you believe is related to security, such as a cross-site scripting vulnerability, do not report it in the standard issue queue. Instead, report it to the security team. See the [*Drupal.org page "How to report a security issue"*](#) for details.

Related topics

[*Bagian 14.2, "Getting Support"*](#)

Additional resources

- [*"Code of Conduct" on Drupal.org*](#)
- [*"Event Calendar" on groups.drupal.org or drupical.com*](#)
- [*"Where is the Drupal Community?" page on Drupal.org*](#)
- [*groups.drupal.org*](#)
- [*"IRC" page on Drupal.org*](#)
- [*"Ways to get involved" page on Drupal.org*](#)
- [*Drupal.org page "Why get involved?"*](#)
- [*Drupal.org page "Contributor tasks"*](#)
- [*Drupal.org page "Use the issue queue"*](#)
- [*Drupal.org page "Reporting a problem"*](#)
- [*Drupal.org page "How to report a security issue"*](#)

Attributions

Written by [*Jennifer Hodgdon*](#), and [*Joe Shindelar*](#) at [*Drupalize.Me*](#).

14.2. Getting Support

Prerequisite knowledge

- [*Bagian 1.6, "Concept: The Drupal Project"*](#)

- [*Bagian 14.1, "Connecting with the Community"*](#)

Where can you find support?

The Drupal project is open-source, so if you have questions about or problems with the software, your options for finding answers and fixes are somewhat different from what they would be for commercial software.

There are several options for free support provided by community volunteers. First, some IRC channels, local groups, and language communities encourage support questions, through on-line chat, in-person meetings, or websites (find out more about these on [*Bagian 14.1, "Connecting with the Community"*](#)).

Second, some (but not all) contributed module, distribution, and theme projects encourage you to post support requests in issues. Generally, projects that have a very large number of users do not allow support requests in their issues (Drupal Core is in that category), while projects with a smaller number of users welcome the occasional support question. Be respectful of developer time and read the documentation for the project before posting a question in an issue. On the other hand, all projects encourage you to use issues to report problems and bugs; see [*Bagian 14.1, "Connecting with the Community"*](#) to learn more about that.

In addition to those resources, the following sites provide free forums where volunteers answer support questions about Drupal:

- The [*Drupal.org Forums*](#)
- [*"Drupal Answers" on StackExchange*](#)

If you prefer to pay for support (presumably in exchange for more extensive service or better availability), you can find service providers in the [*"Drupal Marketplace" on Drupal.org*](#).

Related topics

- [*Bagian 14.3, "Learning More"*](#)
- [*Bagian 14.1, "Connecting with the Community"*](#)

Additional resources

- [*"Support" page on Drupal.org*](#)
- [*"Drupal Marketplace" on Drupal.org*](#)
- The [*Drupal.org Forums*](#)
- [*"Drupal Answers" on StackExchange*](#)

Attributions

Written by [*Jennifer Hodgdon*](#).

14.3. Learning More

Prerequisite knowledge

- [*Bagian 1.6, "Concept: The Drupal Project"*](#)
- [*Bagian 14.1, "Connecting with the Community"*](#)

Where can you go to learn more?

The following resources should prove useful to you, as you continue to advance your skills:

[*"Drupal Community Documentation" section on Drupal.org*](#)

Wiki-like documentation contributed by the Drupal community, ranging from basic to advanced, about all aspects of Drupal (site building and administration, theming, development, and contributed modules).

[*api.drupal.org*](#)

Reference documentation for programmers about the Drupal API.

[*"Drupal Planet" page on Drupal.org*](#)

An aggregation of blog posts about Drupal from around the web, posted by blogging members of the Drupal community who have applied for inclusion. Topics range from programming to site building to Drupal news.

Groups, events, and meetups

See [*Bagian 14.1, "Connecting with the Community"*](#) for more about topical, regional, and language groups, as well as local, regional, and international events.

[*"Training Marketplace" page on Drupal.org*](#)

The Marketplace lists paid training providers. For free training, check whether events include training sessions; there are also free or very low-cost training events listed on the [*"Global Training Days" page on Drupal.org*](#).

Support sites

See [*Bagian 14.2, "Getting Support"*](#) to locate support forums; searching them can be useful for learning about specific topics.

Attributions

Written by [*Jennifer Hodgdon*](#), and [*Joe Shindelar*](#) at [*Drupalize.Me*](#).

Glosarium

Ajax

A web technology used to exchange data with a server to dynamically update parts of a web page (for example, forms) without needing entire page reloads.

Alias

A user-friendly name to replace the internal [path](#) that the system assigns to a URL on the site. For example, you might assign an alias of `/about` to the About page on your site, to replace the internal path `/node/5`. This would give the page a URL of `http://example.com/about` instead of `http://example.com/node/5`. See [Bagian 5.1, "Concept: Paths, Aliases, and URLs"](#) for more information.

Anonymous

A person ([user](#)) interacting with the site who is not logged in. See [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#) for more information.

Block

A chunk of [content](#) (text, images, links, etc.) that can be displayed on a page of a site. Blocks are displayed in [regions](#). See [Bagian 8.1, "Concept: Blocks"](#) for more information.

Breakpoint

Breakpoints are used to separate the height or width of browser screens, printers, and other media output types into steps. A [responsive](#) site adjusts its presentation at these breakpoints. See [Bagian 6.14, "Concept: Responsive Image Styles"](#) for more information.

Bundle

Synonym for [Entity subtype](#).

Cache

The site's internal cache stores the output of time-consuming calculations, such as computing output for an HTML page request, and then retrieves them instead of recalculating the next time they are needed. External caching systems can also be used on the web server to speed up a site's response. See [Bagian 12.1, "Concept: Cache"](#) for more information on the internal cache.

CMS

Acronym for [Content Management System](#).

Configuration

Information about your site that is not [content](#), and is meant to be more permanent than [state](#) information, such as the name of your site, the [content types](#) and [views](#) you have defined, etc. See [Bagian 1.5, "Concept: Types of Data"](#) for more information.

Content

Information meant to be displayed on your site, such as text, images, downloads, etc. See also [Configuration](#) and [State](#). See [Bagian 1.5, "Concept: Types of Data"](#) for more information.

Content item

An item of [content](#) that is typically meant to be displayed as the main content of a page on your site. This is an [entity type](#). See [Bagian 2.3, "Concept: Content Entities and Fields"](#) for more information.

Content Management System (CMS)

A collection of tools designed to allow the creation, modification, organization, search, retrieval and removal of information on a website. See [Bagian 1.1, "Concept: Drupal as a Content Management System"](#) for more information.

Content type

An [entity subtype](#) for the [content item entity type](#). Each content type is used for some particular purpose on the site, and each has its own fields. For example, a site for a farmers market might have a content type for simple pages, and another for a vendor listing page. See [Bagian 2.3, "Concept: Content Entities and Fields"](#) for more information.

Contextual link

A link to an administrative page for editing or configuring a feature of the site, shown in the context where that feature is displayed. Example: a link to configure a [menu](#) that is shown when you hover your mouse over the menu. See [Bagian 4.1, "Concept: Administrative Overview"](#) for more information.

Contributed

[Modules](#), [themes](#), and [distributions](#) that are not part of the [Drupal core](#) download, and that can be downloaded separately from the [Drupal.org](#) website.

Cron

On some operating systems, *cron* is a command scheduler application that executes commands or scripts periodically. Your site defines periodic tasks, also known as cron tasks, that need to be triggered either by an operating system cron scheduler, or internally. See [Bagian 13.1, "Concept: Cron"](#) for more information.

Distribution

A single download that provides a shortcut for setting up a specific type of site, such as a website for a club or for e-commerce. A distribution contains [Drupal core](#), along with [contributed modules](#) and/or [themes](#); many distributions also pre-configure the site or even create sample content upon installation. See [Bagian 1.4, "Concept: Distributions"](#) for more information.

Drupal core

The files, themes, profiles, and modules included with the standard project software download. See [Bagian 1.1, "Concept: Drupal as a Content Management System"](#) for more information.

Entity

An item of either [content](#) or [configuration](#) data, although in common usage, the term often refers to content entities. Examples include [content items](#), custom [blocks](#), [taxonomy](#)

[terms](#), and definitions of [content types](#); the first three are content entities, and the last is a configuration entity. See also [Entity type](#), [Entity subtype](#), and [Field](#). See [Bagian 2.3, "Concept: Content Entities and Fields"](#) for more information.

Entity subtype

Within a [content entity type](#), a grouping of entities that share the same [fields](#). For example, within the [content item](#) entity type, a farmers market site might have subtypes (known as [content types](#)) for static pages and vendor pages, each with its own group of fields. You may also see the term *bundle* used (especially in programmer documentation) as a synonym of entity subtype. See [Bagian 2.3, "Concept: Content Entities and Fields"](#) for more information.

Entity type

The overall type of an [entity](#); in common usage, it is only applied to a [content](#) entity. Examples include [content types](#), [taxonomy terms](#), and custom [blocks](#). See [Bagian 2.3, "Concept: Content Entities and Fields"](#) for more information.

Field

Data of a certain type that is attached to a [content entity](#). For instance, on a farmers market site's vendor content type, you might have fields for an image, the vendor description, and a [taxonomy term](#). See [Bagian 2.3, "Concept: Content Entities and Fields"](#) for more information.

Field bundle

Synonym for [Entity subtype](#).

Field formatter

[Configuration](#) that defines how the data in a [field](#) is displayed. For example, a text field could be displayed with a prefix and/or suffix, and it could have its HTML tags stripped out or limited. See also [View mode](#) and [Field widget](#). See [Bagian 6.10, "Concept: View Modes and Formatters"](#) for more information.

Field widget

[Configuration](#) that defines how someone can enter or edit data for a [field](#) on a data entry form. For example, a text field could use a single-line or multi-line entry box, and there could be a setting for the size of the box. See also [Field formatter](#). See [Bagian 6.8, "Concept: Forms and Widgets"](#) for more information.

Formatter

See [Field formatter](#).

FOSS

Acronym for *Free and Open Source Software*, meaning software that is developed by a community of people and released under a non-commercial license. See also [GPL](#). See [Bagian 1.6, "Concept: The Drupal Project"](#) for more information.

GPL

Acronym for the *GNU General Public License*, a non-commercial software license. All software downloaded from the [Drupal.org](#) website is licensed under the "[GNU General Public License, version 2](#)". See also [FOSS](#). See [Bagian 1.7, "Concept: Drupal Licensing"](#) for more information.

Image style

A set of processing steps that transform a base image into a new image; typical processing includes scaling and cropping. See [Bagian 6.12, "Concept: Image Styles"](#) for more information.

LAMP

Acronym for *Linux, Apache, MySQL, and PHP*: the software on the web server that the scripts commonly run on (although it can use other operating systems, web servers, and databases). See [Bagian 3.1, "Concept: Server Requirements"](#) for more information.

Log

A list of recorded events on the site, such as usage data, performance data, errors, warnings, and operational information. See [Bagian 12.4, "Concept: Log"](#) for more information.

Menu

A set of links used for navigation on a site, which may be arranged in a hierarchy. See [Bagian 5.6, "Concept: Menu"](#) for more information.

Module

Software (usually PHP, JavaScript, and/or CSS) that extends site features and adds functionality. The Drupal project distinguishes between [core](#) and [contributed](#) modules. See [Bagian 1.2, "Concept: Modules"](#) for more information.

Path

The unique, last part of the internal URL that the system assigns to a page on the site, which can be a visitor-facing page or an administrative page. For example, the internal URL for the About page on your site might be <http://example.com/node/5>, and in this case, the path is *node/5*. See also [Alias](#). See [Bagian 5.1, "Concept: Paths, Aliases, and URLs"](#) for more information.

Permission

The ability to perform some action on the site, such as editing a particular type of [content](#), or viewing user profiles. See also [Role](#). See [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#) for more information.

Reference field

A [field](#) that represents a relationship between an [entity](#) and one or more other entities, which may be the same [entity type](#) or a different type. For example, on a farmers market site, a recipe content item might have a reference field to the vendor (also a content item) that posted the recipe. [Taxonomy term](#) fields are also reference fields. See [Bagian 6.4, "Concept: Reference Fields"](#) for more information.

Region

A defined area of a page where [content](#) can be placed, such as the header, footer, main content area, left sidebar, etc. Regions are defined by [themes](#), and the content displayed in each region is contained in [blocks](#). See [Bagian 2.1, "Concept: Regions in a Theme"](#) for more information.

Responsive

A site or [theme](#) is said to be responsive if it adjusts its presentation in response to the size of the browser screen, printer, or other media output type. See also [Breakpoint](#). See [Bagian 6.14, "Concept: Responsive Image Styles"](#) for more information.

Revision

A record of the past or present state of a [content entity](#), as it is edited over time. See [Bagian 2.6, "Concept: Editorial Workflow"](#) for more information.

Role

A named set of [permissions](#) that can be applied to a [user account](#). See [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#) for more information.

Security update

An [update](#) that fixes a security-related bug, such as a hacking vulnerability. See [Bagian 13.3, "Concept: Security and Regular Updates"](#) for more information.

State

Information of a temporary nature about the current state of your site, such as the time when [cron](#) was last run, etc. See also [Content](#) and [Configuration](#). See [Bagian 1.5, "Concept: Types of Data"](#) for more information.

Taxonomy

The process of classifying [content](#). See [Bagian 6.5, "Concept: Taxonomy"](#) for more information.

Taxonomy term

A term used to classify [content](#), such as a tag or a category. See also [Vocabulary](#). See [Bagian 6.5, "Concept: Taxonomy"](#) for more information.

Text format

[Configuration](#) that defines the processing that happens to user-entered text before it is shown in the browser. This might include stripping or limiting HTML tags, or turning URLs into links. See [Bagian 6.15, "Concept: Text Formats and Editors"](#) for more information.

Theme

Software and asset files (images, CSS, PHP code, and/or templates) that determine the style and layout of the site. The Drupal project distinguishes between [core](#) and [contributed](#) themes. See [Bagian 1.3, "Concept: Themes"](#) for more information.

UI

Acronym for [User Interface](#).

Update

A newer version of your site's software, either [Drupal core](#) or a [module](#) or [theme](#). See also [Security update](#). See [Bagian 13.3, "Concept: Security and Regular Updates"](#) for more information.

User

A person interacting with the site, either logged-in or [anonymous](#). See [Bagian 7.1, "Concept: Users, Roles, and Permissions"](#) for more information.

User interface

The text, styles, and images that are visible on a site, separated logically into the user interface for site visitors and the administrative user interface.

User one (User 1)

The initial [user](#) account that is created when you install the site (whose ID number is 1). It automatically has all [permissions](#), even if it is not assigned an administrative [role](#). See [Bagian 7.2, "Concept: The User 1 Account"](#) for more information.

View

A formatted listing of data; typically, the data comes from [content entities](#). For example, on a farmers market site, you might create a [content item](#) for each vendor. You could then make view that generates a listing page that shows a thumbnail image and short description of each vendor, linking to the full-page content item. Using the same data, you could also make a view that generates a new vendors block, which would show information from the most recently added vendors. See [Bagian 2.4, "Concept: Modular Content"](#) for more information.

View mode

A set of [field formatter configuration](#) for all of the [fields](#) of a [content entity](#), some of which may be hidden. Each [entity subtype](#) can have one or more view modes defined; for example, [content types](#) typically have *Full* and *Teaser* view modes, where the *Teaser* view mode displays fewer or trimmed-down fields. See [Bagian 6.10, "Concept: View Modes and Formatters"](#) for more information.

Vocabulary

A group of [taxonomy terms](#) to choose from when classifying [content](#) in a particular way, such as the list of all of the vendor categories on a farmers market site. Technically, vocabularies are the [entity subtype](#) for the taxonomy term [entity type](#). See [Bagian 6.5, "Concept: Taxonomy"](#) for more information.

Widget

See [Field widget](#).

Wizard

A web form that allows you to fill in a few values, and creates something with sensible defaults based on the values you chose. For example, there are wizards for creating [views](#) of different types. See [Bagian 9.3, "Creating a Content List View"](#) for more information.

WYSIWYG

Acronym for *What You See is What You Get*, meaning a method for editing [content](#) where what you see on the editing screen closely resembles the final product. See [Bagian 6.16, "Configuring Text Formats and Editors"](#) for more information.

Indeks

A

Account setting
 configuring, 41
Admin Toolbar module
 downloading, 149
 installing, 149
Administration
 overview, 32
Administrative interface
 using to update the core software, 184
Administrative menu
 overview, 32, 54
Administrative user account
 overview, 98
Ajax
 definition, 196
Alias
 definition, 196
 overview, 47
Anonymous
 definition, 196
Anonymous user
 overview, 98
Apache web server
 version requirements, 20
Appearance menu entry
 administrative menu, 32
Attributions for this document, 212
Authenticated user
 overview, 98
Author
 assigning, 109
Automated Cron module
 configuring, 179
 overview, 179

B

Backup
 overview, 175
Bartik theme
 configuring, 43
 regions in, 10
Basic page
 creating, 48
Block

creating, 112
creating from a view, 127
definition, 196
entity type, 13
overview, 112
placing in a region, 114
Breadcrumb region
 overview, 10
Breakpoint
 definition, 196
 overview, 94
Bug
 reporting, 182, 192
Bundle
 definition, 196

C

Cache
 clearing, 173
 definition, 196
 overview, 172
 rebuilding, 173
Chatting online, 192, 194
CKEditor module
 configuring, 95
CKEditor text editor
 default assignment to text format, 95
CMS (Content Management System)
 definition, 196
 overview, 1
Coder tool
 overview, 21
Color scheme
 configuring, 43
Comment entity type
 overview, 13
Comment module
 entity type, 13
Community
 connecting with, 192
Composer tool
 overview, 21
 using to download modules, 163
 using to download the core software, 163
 using to download themes, 163
Composite page
 creating with modular content, 15
Configuration
 as a type of data, 6

- copying, 166
- definition, 196
- deploying, 166
- synchronizing between development and live site, 169
- translating, 132, 139
- Configuration Manager module
 - using to synchronize the configuration between a development and live site, 169
- Configuration menu entry
 - administrative menu, 32
- Configuration Translation module
 - installing, 133
 - overview, 132
- Configuring
 - content translation, 134
 - home page, 53
 - site, 35
 - text format, 95
- Contact form entity type
 - overview, 13
- Contact module
 - entity type, 13
- Content
 - adding to menu while editing, 56
 - as a type of data, 6
 - assigning author, 109
 - backing up, 175
 - definition, 196
 - displaying, 86
 - editing, 50, 52
 - editing entry form, 82
 - entity type, 13
 - finding, 50
 - modular, 15
 - translating, 132, 138
 - translation, 134
- Content attributions for this document, 212
- Content display
 - changing, 86
 - managing, 86
- Content editing form
 - overview, 82
- Content entity type
 - overview, 13
- Content item
 - creating, 48
 - definition, 197
- Content list view
 - creating, 119
- Content Management System (CMS)
 - definition, 196, 197
 - overview, 1
- Content menu entry
 - administrative menu, 32
- Content reference field
 - adding, 78
 - overview, 71
- Content region
 - overview, 10
- Content structure
 - planning, 16
- Content Translation module
 - installing, 133
 - overview, 132
- Content type
 - adding, 61
 - adding field to, 66
 - definition, 197
 - deleting, 65
- Contextual filters part of view
 - overview, 118
- Contextual link
 - definition, 197
 - overview, 32
- Contributed
 - definition, 197
- Contributed module
 - definition, 197
 - downloading, 149, 158, 163
 - evaluating, 143
 - finding, 143
 - installing, 149
 - overview, 3
 - updating, 186
- Contributed theme
 - definition, 197
 - downloading, 155, 158, 163
 - enabling, 155
 - evaluating, 152
 - finding, 152
 - installing, 155
 - updating, 188
- Contributing to the project, 192
- Copyright for this document, vi, 212
- Core module
 - overview, 3

- Core software
 - installation requirements, 20
 - installing, 26
 - updating, 184
- Core theme
 - overview, 4
- Country setting
 - configuring, 35
- Cron command scheduler
 - definition, 197
- Cron task
 - configuring, 179
 - definition, 197
 - overview, 179
- Cross-site scripting (XSS)
 - preventing, 94
- Custom block
 - creating, 112
 - entity type, 13
- Custom block module
 - entity type, 13
- Custom menu
 - overview, 54
- Custom theme
 - enabling, 155
 - installing, 155
 - overview, 4
- D**
 - Data type
 - configuration, 6
 - content, 6
 - overview, 6
 - session, 6
 - state, 6
 - Database
 - backing up, 175
 - configuring during the installation process, 26
 - installation requirements, 20
 - Deployment workflow
 - overview, 160
 - Devel tool
 - overview, 21
 - Development site
 - making, 161
 - overview, 160
 - synchronizing with live site, 169
 - Disk space
- installation requirements, 20
- Display part of view
 - overview, 118
- Distribution
 - definition, 197
 - full-featured, 5
 - overview, 5
 - quick-start, 5
- Documentation
 - about Drupal, 195
 - about the Drupal API, 195
 - available on drupal.org, 195
 - contributed by the Drupal Community, 195
 - Drupal Planet blog posts, 195
 - for programmers, 195
- Downloading
 - core software, 23, 163
 - module, 149, 163
 - module or theme files manually, 158
 - theme, 155, 163
- Drupal Association
 - overview, 7
- Drupal Console tool
 - overview, 21
- Drupal content management system
 - overview, 1
 - server requirements, 1
- Drupal core
 - definition, 197
 - installing, 26
 - overview, 1
 - preparing install, 23
- Drupal licensing
 - overview, 8
- Drupal project
 - connecting with, 192
 - overview, 7
- Drupal training
 - resources for, 195
- Drupal user group
 - finding, 192
- Drupal.org website
 - downloading and installing module from, 149
 - downloading and installing theme from, 155
 - downloading modules and themes from, 1
 - finding and evaluating modules on, 143

- finding and evaluating themes on, 152
- Drush tool**
- overview, 21
 - using to download and install module, 149
 - using to download and install theme, 155
 - using to install the core software, 23
 - using to uninstall module, 39
 - using to update module, 186
 - using to update the core software, 184
 - using to update theme, 188
- Dynamic Page Cache module**
- overview, 172
- E**
- Editing content**, 50, 52
- Editor**
- configuring, 95
- Editor module**
- configuring, 95
- Editorial Workflow**
- overview, 18
- Effect**
- image, 90, 90
- Email address**
- configuring, 35
- Enabling**
- module, 149
 - theme, 155
- Entity**
- definition, 197
 - overview, 13
- Entity reference field**
- adding, 78
- Entity subtype**
- definition, 198
 - overview, 13
- Entity type**
- definition, 198
 - overview, 13
- Entity view mode**
- overview, 86
- Error**
- log report, 176
- Error page**
- configuring, 35
- Event**
- Drupal, 192
- Extend menu entry**
- administrative menu, 32
- F**
- Farmers market**
- usage scenario for this document, xi
- Feature**
- copying, 166
 - deploying, 166
- Featured regions**
- overview, 10
- Field**
- adding to content type, 66
 - content reference, 71
 - definition, 198
 - for adding references, 78
 - overview, 13, 82
 - reference, 71
 - taxonomy term reference, 71
 - user reference, 71
- Field bundle**
- definition, 198
- Field formatter**
- definition, 198
 - overview, 86
- Field module**
- overview, 3
- Field UI module**
- overview, 3
- Field widget**
- definition, 198
- Fields part of view**
- overview, 118
- File**
- backing up, 175
- File module**
- entity type, 13
- Filter criteria part of view**
- overview, 118
- Filter module**
- configuring, 95
- Firebug tool**
- overview, 21
- First day of week**
- configuring, 35
- Footer menu**
- overview, 54
- Footer regions**
- overview, 10
- Form**
- overview, 82

- Form entity type
 - overview, 13
- Format part of view
 - overview, 118
- Formatter
 - definition, 198
 - overview, 86
- Forum
 - user support, 194
- FOSS (Free and Open Source Software)
 - definition, 198
 - overview, 7
- Free and Open Source Software (FOSS)
 - definition, 198
 - overview, 7
- Front page
 - configuring, 35, 53
- Full editor, 52
- Full-featured distribution
 - overview, 5
- Functionality
 - extending, 149
- G**
 - Git tool
 - overview, 21
 - Glossary, 196
 - GNU General Public License
 - definition, 198
 - overview, 8
 - GPL (General Public License or GNU General Public License)
 - definition, 198
 - overview, 8
 - Group
 - finding, 192
 - Guiding scenario for this document (farmers market), xi
- H**
 - Header region
 - overview, 10
 - Help menu entry
 - administrative menu, 32
 - Help region
 - overview, 10
 - Hiawatha web server
 - version requirements, 20
 - Highlighted region
- overview, 10
- Home page
 - configuring, 53
 - creating, 48
- HTML5 picture tag
 - and responsive images, 94
- I**
 - Image
 - resizing, 90
 - Image effect
 - overview, 90
 - Image field
 - adding, 66
 - Image module
 - creating a view, 119
 - Image style
 - creating, 90
 - definition, 198
 - overview, 90
 - responsive, 94
 - In-place editor, 52
 - Installation process, 26
 - Installation profile, 26
 - Installation requirements
 - database, 20
 - disk space, 20
 - overview, 20
 - PHP programming language, 20
 - web server, 20
 - Installer tool
 - running, 26
 - Installing
 - core software, 20, 23
 - development site, 160
 - module, 149
 - staging site, 160
 - theme, 155
 - Interest group
 - finding, 192
 - Interface Translation module
 - installing, 133
 - overview, 132
 - Internal Page Cache module
 - overview, 172
 - Internet Relay Chat (IRC)
 - using to chat online, 192, 194
 - IRC (Internet Relay Chat)
 - using to chat online, 192, 194

L

Label
 translating, 139

LAMP (Linux Apache MySQL PHP)
 definition, 199

Language
 adding, 133

Language module
 installing, 133
 overview, 132

Layout
 planning, 11

Learning resource
 link to, 195

Legal
 overview, 8

Licensing
 overview, 8

Linux Apache MySQL PHP (LAMP)
 definition, 199

Listing content
 using Views module to, 119

Live site
 synchronizing with development site, 169

Locale setting
 configuring, 35

Log
 definition, 199
 overview, 176

Logo image
 configuring, 43

M

Maintenance mode
 disabling, 147
 enabling, 147
 overview, 147

Manage administrative menu
 overview, 32

Manually downloading module or theme files
 overview, 158

Menu
 adding a link to page, 56
 administrative, 54
 custom, 54
 definition, 199
 footer, 54

main, 54
overview, 54
user account, 54

Menu items
 reordering, 58

Menu region
 overview, 10

Menu UI module
 overview, 3

Metadata
 creating, 73

Microsoft IIS web server
 version requirements, 20

Modular content
 overview, 15

Module
 Admin Toolbar, 149
 CKEditor, 95
 Comment, 13
 Configuration Manager, 169
 Configuration Translation, 132, 133
 Contact, 13
 Content Translation, 132, 133
 contributed, 3, 149
 core, 3
 custom, 149
 Custom Block, 13
 definition, 199
 downloading, 149, 158, 163
 Dynamic Page Cache, 172
 Editor, 95
 enabling, 37, 149
 evaluating, 143
 Field, 3
 Field UI, 3
 File, 13
 Filter, 95
 finding, 143
 Image, 119
 installing, 37, 149
 Interface Translation, 132, 133
 Internal Page Cache, 172
 Language, 132, 133
 Menu UI, 3
 Node, 3, 13
 overview, 3
 Taxonomy, 13
 uninstalling unused, 39
 Update Manager, 149, 155

- updating, 186
 - User, 3, 13
 - Views, 3, 119, 124, 127
 - Views UI, 3
 - Multilingual modules
 - enabling, 133
 - installing, 133
 - MySQL database
 - version requirements, 20
 - N**
 - Navigation
 - changing the order of menu items, 58
 - linking menu to page, 56
 - Nginx web server
 - version requirements, 20
 - Node module
 - entity type, 13
 - overview, 3
 - P**
 - Page
 - adding to menu while editing, 56
 - composite, 15
 - Page cache
 - overview, 172
 - Path
 - definition, 199
 - overview, 47
 - People menu entry
 - administrative menu, 32
 - Performance
 - improving, 39
 - log report, 176
 - status report, 177
 - Permission
 - changing, 105
 - changing role, 107
 - definition, 199
 - denying, 105
 - granting, 105
 - overview, 98
 - PHP programming language
 - version requirements, 20
 - PHP version
 - status report, 177
 - Planning
 - site layout, 11
 - PostgreSQL database
 - version requirements, 20
 - Primary menu region
 - overview, 10
 - Profile
 - installation, 26
 - Published flag
 - overview, 18
- Q**
 - Quick editor, 52
 - Quick-start distribution
 - overview, 5
- R**
 - Reference field
 - adding, 78
 - content, 71
 - definition, 199
 - overview, 71
 - taxonomy term, 71
 - user, 71
 - Region
 - breadcrumb, 10
 - content, 10
 - definition, 199
 - featured, 10
 - footer, 10
 - header, 10
 - help, 10
 - highlighted, 10
 - menu, 10
 - overview, 10
 - placing a block in, 114
 - primary menu, 10
 - secondary menu, 10
 - sidebar, 10
 - Regional setting
 - configuring, 35
 - Relationships part of view
 - overview, 118
 - Report
 - Recent log messages, 176
 - Reports menu entry
 - administrative menu, 32
 - Resource
 - documentation and training, 195
 - Responsive
 - definition, 199
 - Responsive image style

- overview, 94
- Revision**
 - definition, 199
 - overview, 18
- Role**
 - administrator, 100
 - anonymous user, 100
 - authenticated user, 100
 - changing, 107
 - changing permission, 105
 - creating, 100
 - definition, 200
 - overview, 98
- S**
 - Scenario for this document (farmers market)
 - overview, xi
 - Secondary menu region
 - overview, 10
 - Security
 - assigning permission, 105
 - cross-site scripting, 94
 - managing user accounts, 41
 - overview, 182
 - status report, 177
 - text format, 94
 - user account settings, 41
 - user one account, 99
 - Security announcement
 - subscribing to by email, 183
 - subscribing to on Twitter, 183
 - Security bug
 - reporting, 182
 - Security team
 - overview, 182
 - Security update
 - applying, 186, 188
 - definition, 200
 - keeping track of, 183
 - Server information
 - status report, 177
 - Session
 - as a type of data, 6
 - Sidebar regions
 - overview, 10
 - Site email address
 - configuring, 35
 - Site information
 - status report, 177
 - Site layout
 - planning, 11
 - Site name
 - configuring, 35
 - Site plan
 - content structure, 16
 - Site slogan
 - configuring, 35
 - Site tagline
 - configuring, 35
 - Slogan
 - configuring, 35
 - Software dependencies
 - managing, 163
 - Software version
 - status report, 177
 - Sort criteria part of view
 - overview, 118
 - SQLLight database
 - version requirements, 20
 - Staging site
 - making, 161
 - overview, 160
 - State
 - as a type of data, 6
 - definition, 200
 - Status report
 - overview, 177
 - Structure menu entry
 - administrative menu, 32
 - Style
 - image, 90, 90
 - Support
 - finding, 194
- T**
 - Tagline
 - configuring, 35
 - Taxonomy
 - creating, 73
 - definition, 200
 - overview, 72
 - Taxonomy module
 - entity type, 13
 - Taxonomy term
 - definition, 200
 - overview, 13
 - Taxonomy term reference field
 - adding, 78

- overview, 71
- Term (taxonomy)**
 - fixed list, 72
 - free tagging, 72
 - overview, 72
- Term list**
 - creating, 73
- Terminology (Glossary), 196**
- Text format**
 - configuring, 95
 - definition, 200
 - overview, 94
- Theme**
 - configuring, 43
 - contributed, 155
 - core, 4
 - custom, 4, 155
 - definition, 200
 - downloading, 155, 158, 163
 - enabling, 155
 - evaluating, 152
 - finding, 152
 - installing, 155
 - overview, 4
 - regions in, 10
 - third-party, 4
 - updating, 188
- Third-party theme**
 - overview, 4
- Time zone setting**
 - configuring, 35
- Tool**
 - Coder, 21
 - Composer, 21
 - Devel, 21
 - drupal Console, 21
 - Drush, 21
 - Firebug, 21
 - Git, 21
- Toolbar**
 - overview, 32
- Training**
 - Global Training Days, 195
 - resource, 195
 - Training Marketplace, 195
- Translating**
 - configuration, 139
 - content, 138
 - label, 139
- overview, 132
- view, 139
- Troubleshooting**
 - log report, 176
 - status report, 177
- U**
- UI (User Interface)**
 - definition, 200
- Uniform Resource Locator (URL)**
 - overview, 47
- Uninstalling**
 - unused modules, 39
- Unpublished flag**
 - overview, 18
- Update**
 - definition, 200
 - keeping track of, 183
 - overview, 182
 - status report, 177
- Update Manager module**
 - overview, 183
 - using to install module, 149
 - using to install theme, 155
- Update status**
 - status report, 177
- URL (Uniform Resource Locator)**
 - overview, 47
- URL alias**
 - overview, 47
- URL field**
 - adding, 66
- Usage scenario for this document (farmers market), xi**
- User**
 - administrative, 99
 - configuring account setting, 41
 - creating account, 102
 - definition, 200
 - overview, 98
 - root, 99
 - user one, 99
- User account menu**
 - overview, 54
- User group**
 - finding, 192
- User interface**
 - translating, 132
- User Interface (UI)**

definition, 200

User module

entity type, 13

overview, 3

User one

definition, 200

User profile entity type

overview, 13

User reference field

adding, 78

overview, 71

User role

changing, 107

creating, 100

overview, 98

User-entered content

filtering, 94

V

View

adding block display to, 127

and modular content, 15

contextual filters part, 118

creating, 119

definition, 201

display part, 118

duplicating, 124

fields part, 118

filter criteria part, 118

format part, 118

overview, 117

parts of, 118

relationships part, 118

sort criteria part, 118

translating, 139

View mode

definition, 201

overview, 86

Views module

adding to a view, 127

creating a view, 119

duplicating a view, 124

overview, 3, 15, 117

Views UI module

overview, 3

Vocabulary

creating, 73

definition, 201

overview, 13, 72

W

Web installer

running, 26

using to install the core software, 23

Web server

installation requirements, 20

What You See Is What You Get (WYSIWYG)

configuring editor, 95

definition, 201

Widget

definition, 198, 201

overview, 82

Wizard

definition, 201

WYSIWYG (What You See Is What You Get)

configuring editor, 95

definition, 201

X

XSS (Cross-site scripting)

preventing, 94

Lampiran A. Lampiran

A.1. Guide-Wide Attributions

This guide was written by contributors to the Drupal open-source project, and is licensed under the [CC BY-SA 2.0](#) license. See [Bagian i.1, "Copyright"](#) for more information. Details about the contributors for guide-wide tasks are below. For individual topics, the attributions for writing, editing, and/or translating are at the end of each topic.

Project coordination of original (English, Drupal 8) text

Writing of the initial version of this guide was coordinated by the following people:

- [Amber Matz](#) - Volunteer coordination and project management
- [Jennifer Hodgdon](#) - Content oversight
- [Joe Shindeler](#) - Content oversight
- [Antje Lorch](#) - Content oversight

Writing, editing, and testing of original text

The following people contributed to guide-wide writing:

- The glossary was written and adapted by [Jennifer Hodgdon](#), with parts from "[Glossary](#)", "[Overview of Configuration \(vs. other types of information\)](#)", and "[Working with breakpoints in Drupal 8](#)", each copyright 2000-2016 by the individual contributors to the [Drupal Community Documentation](#).
- The index was improved and curated by [Anna van Raaphorst](#).
- Topic summaries were written by [Jojo Alphonso](#) at [Red Crackle](#).

Guide-wide editing was done by the following people:

- [Amber Matz](#) at [Drupalize.Me](#)
- [Anna van Raaphorst](#)
- [Antje Lorch](#)
- [Balu Ertl](#)
- [Boris Doesborg](#)
- [Charles Leverington](#)
- [chirag shah](#)
- [Grant Dunham](#)
- [Jennifer Hodgdon](#)

- [Joe Shindelar](#) at [Drupalize.Me](#)
- [Jojo Alphonso](#) at [Red Crackle](#) — who copy edited the vast majority of the User Guide
- [Kamal Kant Pansari](#) at [Intelliswift](#)
- [Michael Lenahan](#) at [erdfisch](#)
- [Swarad Mokal](#) at [Blisstering Solutions](#)

The content was tested by:

- [Zach Carter](#)
- A workshop organized by the "[Spokane Drupal User Group](#)" on [groups.drupal.org](#), with students from the Web Development Certificate program at [Spokane Community College](#)

Images

Most of the images in the guide are screen captures from the Drupal software, generated by software written by [Jennifer Hodgdon](#).

The word Drupal, the Druplicon logo (which appears as part of several images throughout the guide), and the Drupal wordmark logo (which is part of the cover image), are [registered trademarks of Dries Buytaert](#).

The cover image is a CC BY-SA 2.0 licensed illustration by [Amanda Luker](#) at [Advomatic](#).

The Anytown Farmers Market logo, which appears as part of several images throughout the guide, is a CC BY-SA 2.0 licensed illustration by Justin Harrell at [Drupalize.Me](#).

The honey bee image, which appears as part of several images throughout the guide, is a public domain image by [John Severns](#) at the [English Wikipedia project](#) via [Wikimedia Commons](#).

The farm image, which appears as part of several images throughout the guide, is a public domain image by Xianmin Chang via [Wikimedia Commons](#).

The salad image, which appears as part of several images throughout the guide, is a public domain image by [Yinan Chen](#) via [Wikimedia Commons](#).

The carrot image, which appears as part of several images throughout the guide, is public domain image [K11611-1](#) by Stephen Ausmus at the [Agricultural Research Service of the United States Department of Agriculture](#) via [Wikimedia Commons](#).

Translation into Bahasa Indonesia

The Bahasa Indonesia translation of this guide was coordinated by:

- [Suryanto Rachmat](#)
- [Pratomo Ardianto](#)